MSX DOS

MET DISK BASIC

A. van Utteren

Kluwer Technische Boeken

Aaldrik van Utteren

MSX-DOS met Disk BASIC

Kluwer Technische Boeken B.V.
Deventer - Antwerpen

Omslag: Wim Niessink

ISBN 90 201 1845 5
D/1986/0108/190

1e druk 1986

INHOUDSOPGAVE

WOORD VOORAF

Inmiddels zijn de drie letters MSX niet onbekend meer in de computerwereld. Steeds meer mensen zien de voordelen van de MSX-computer, de standaard onder de huiscomputers. Eén van de voordelen van de MSX-computer is de mogelijkheid een diskdrive als opslagmedium voor uw programma's en data te gebruiken zonder dat u daar een apart besturingssysteem voor nodig heeft. Het besturingssysteem voor uw diskdrive bevindt zich in de interface die in het uitbreidingsslot van uw computer wordt geplaatst. Na het opstarten zult u op het beeldscherm een mededeling krijgen omtrent de versie van Disk BASIC die nu ter beschikking staat.
Daar er inmiddels al een aantal software-leveranciers zijn die programma's leveren in machinetaal zoals, PASCAL, MULTIPLAN enz., was het nodig om een speciaal besturingssysteem voor deze programma's te maken. Men heeft hiervoor het disk-besturingssysteem MSX-DOS ontwikkeld. Straks zullen veel programma's werken onder MSX-DOS, o.a. de hiervoor genoemde programma's.
Dit boek helpt u een weg te vinden door alle commando's die nieuw zijn voor de MSX-computer, zowel voor Disk BASIC als DOS. Omdat het disk-besturingssysteem meer geheugenruimte nodig heeft dan Disk BASIC, dient u voor MSX-DOS een MSX-computer te gebruiken met een geheugencapaciteit van minimaal 64 k RAM.
Bij het gebruik van een diskdrive is het belangrijk om te profiteren van de snelle toegangstijd van dit medium. Vandaar dat er een hoofdstuk wordt gewijd aan het werken met en maken van bestanden.

Mijn dank gaat uit naar de afdeling Home Interactive Systems van PHILIPS Nederland die het benodigde materiaal beschikbaar heeft gesteld en het manuscript van kritische kanttekenin-

gen heeft voorzien teneinde u als lezer een optimaal naslagwerk te kunnen bieden.

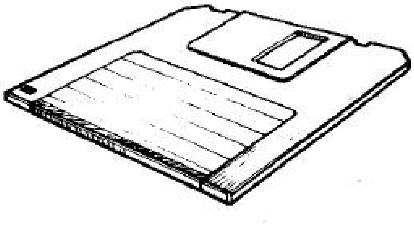
Blaricum, januari 1986 Aaldrik van Utteren

HOOFDSTUK 1

WERKEN MET DISKETTES EN DISKDRIVES

1.1 Wat is een diskette?

Een diskette dient als extern opslagmedium voor computergegevens. In principe kunnen we een diskette vergelijken met de cassetteband van een cassetterecorder. Bij een cassetteband is de drager een smalle, lange band die is voorzien van magnetisch materiaal. Als we de computer opdracht geven de informatie die zich in het geheugen bevindt op band te bewaren, zal de computer die informatie in een zodanige vorm naar de cassetterecorder sturen dat deze het kan vastleggen op het magnetische materiaal. Bij de cassetterecorder staat de opname/weergavekop (in computertaal lees/schrijfkop) stil en beweegt de band langs de kop. Hierdoor komt de informatie als een lange reeks magnetische pulsjes 'achter elkaar' op de cassetteband te staan. Daarom is het nodig om de band steeds terug te spoelen als men gegevens wil lezen die aan het begin van de band staan. Vandaar dat men is gaan zoeken naar een andere vorm die de informatie voor zowel de computer als voor de gebruiker sneller toegankelijk kan maken.

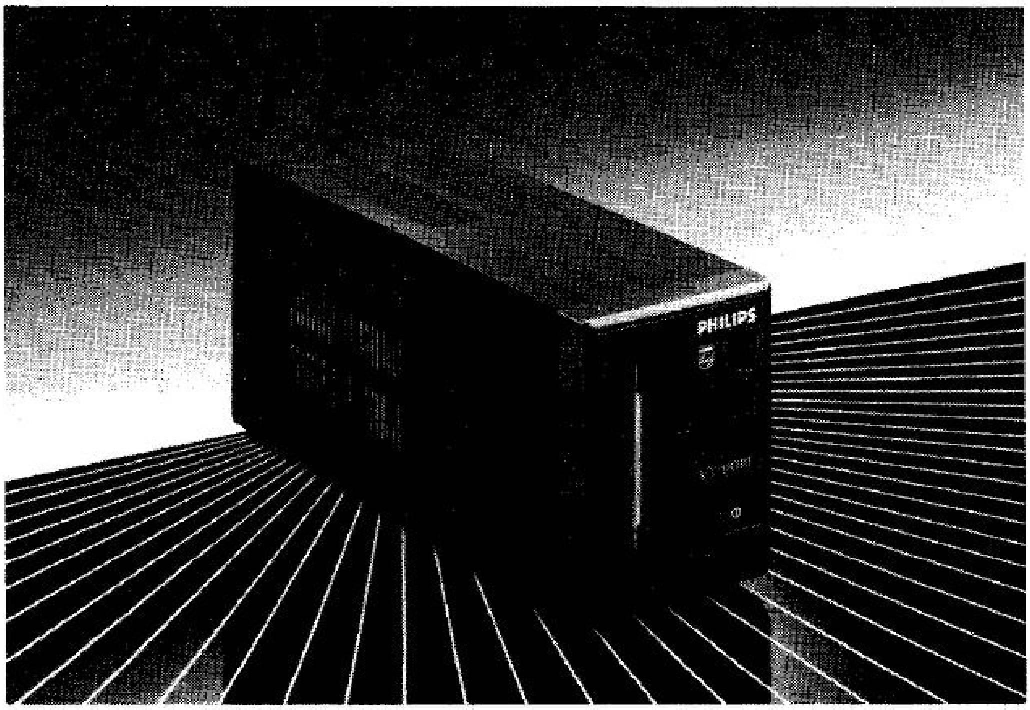


Deze nieuwe vorm werd gevonden in een ronde schijf. Die schijf is aan beide zijden voorzien van een magnetisch gevoelige laag. In principe is zo'n schijf (bij computers 'diskette' genaamd) dan ook aan beide zijden te gebruiken. Vóór een diskette de fabriek verlaat, wordt deze getest op homogeniteit, waarbij men bepaalt of de diskette aan één of aan twee zijden is te gebruiken.
De diskette wordt geplaatst in een speciaal hiervoor ontworpen lees-en-schrijfapparaat (een diskdrive) die de informatie op de diskette kan 'schrijven' en 'teruglezen'. In de diskdrive wordt de diskette met grote snelheid aan het draaien gebracht, u kunt dit vergelijken met de grammofoonplaat of Compact Disc. Een lees/schrijfkop kan zich rechtlijnig langs de straal van de diskette over de magnetische laag bewegen. Omdat de draaisnelheid van de diskette constant is en de beweging van de lees/schrijf-

kop in kleine constante stappen is vastgelegd, weet de diskdrive steeds waar de lees/schrijfkop zich bevindt.
Een nieuwe diskette bevat alleen een magnetische laag waar nog geen informatie op staat. Men kan hem dan ook niet zonder meer gebruiken voor het opslaan van gegevens, omdat de diskdrive die informatie graag wil wegschrijven naar snel terug te vinden delen op de diskette. De informatie moet immers ook weer snel kunnen worden opgezocht en gelezen.
Om deze reden wordt de diskette eerst ingedeeld (formatteren heet dat met een duur woord) met een speciaal commando. Dit indelen of formatteren kan men het beste zien als het aanbrengen van een aantal denkbeeldige cirkels, die op hun beurt weer worden onderverdeeld in sectoren. Door het aanbrengen van merktekens op de diskette 'weet' de lees/schrijfkop exact waar zo'n cirkel zich bevindt en hoe die cirkel is onderverdeeld. Zo'n ronde schijf wordt ook vaak floppy genoemd, terwijl we eveneens namen tegenkomen als floppy disk, schijf en de meest gangbare benaming diskette.
De verscheidenheid in afmetingen van diskettes is soms erg verwarrend, men kent 5 1/4 inch, 3 1/2 inch, 2,8 inch (3 inch genoemd), Winchester en ook nog single side en double side, samen met single en double density. Het is leuk om even met deze technische kreten te hebben kennisgemaakt, maar voor het werken met MSX-DOS of Disk BASIC niet van belang.
MSX beveelt als standaard de 3 1/2 inch-diskette aan, die is voorzien van een hardplastic-behuizing en van een metalen bescherming tegen stof en vuil, die zich over het gat bevindt waar de lees/schrijfkop zich in kan bewegen.
De fabrikanten zijn in principe vrij in de keuze van het toe te passen disketteformaat, doch de meesten leveren diskdrives die zijn afgestemd op de steeds populairder wordende 3 1/2 inch-diskette.

1.2 De diskdrive

In principe kunnen alle voor MSX ontworpen diskdrives worden aangesloten zoals drives voor de 5 1/4 inch- en 3 1/2 inch-diskettes. Voordat u een diskdrive aansluit, altijd de gebruiksaanwijzing van die diskdrive goed doorlezen! Computer altijd uitzetten en daarna pas de interface in het slot (cartridge slot) plaatsen of in een apart verkrijgbaar, zgn. los, uitbreidingsslot. Zo'n uitbreidingsslot bestaat uit een aantal connectoren (sloten) en wordt aangesloten op één van de sloten van de computer, hierdoor heeft u meer uitbreidingsmogelijkheden.
Na het insteken van de interface de diskdrive hierop aansluiten. Dit doet men door de kabel die vastzit aan de interface te verbinden met de aansluiting 'drive A' (computer) aan de achterzijde van de diskdrive. Als u een interface zonder vaste kabel heeft,

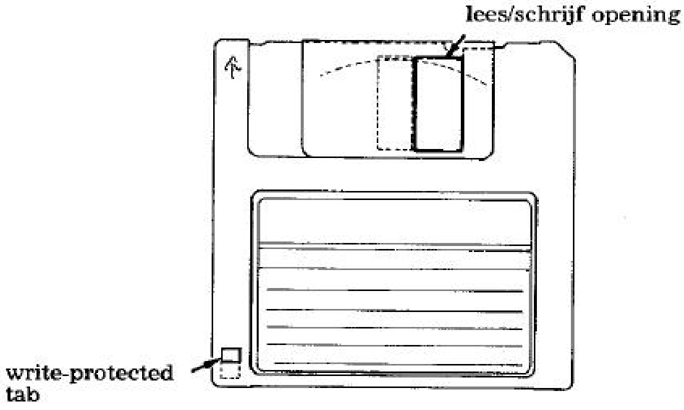


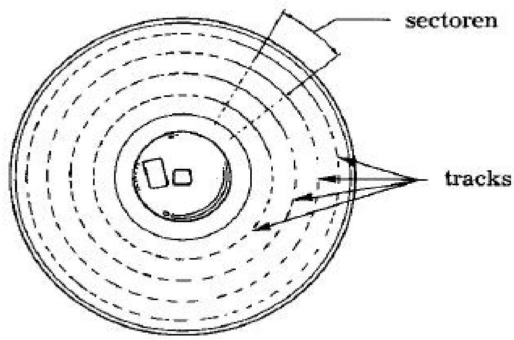
moet u de bijgeleverde 'flatkabel' aansluiten op de interface en op de diskdrive. Aan de achterzijde van de diskdrive bevindt zich nog een aansluiting 'ext. drive' die dient om een tweede diskdrive te kunnen aansluiten. Op deze manier kunt u via de 'eerste' diskdrive een 'tweede' aansluiten die dan samen op één interface zijn aangesloten.
Bij MSX geeft men de diskdrives geen volgnummer maar een letter als kenmerk, de eerste diskdrive geeft men de letter 'A', de volgende 'B' enz. tot 'D'. Zodoende kan men maximaal vier diskdrives aansluiten op de computer, waarbij wel geldt dat u twee interfaces moet gebruiken. Meer dan twee diskdrives op één interface is niet mogelijk. Bij gebruik van één interface maakt het niet uit welk slot u gebruikt voor de interface; als de computer bij het aanzetten één diskdrive-interface ziet, gaat hij ervan uit dat diskdrive A en B kunnen zijn aangesloten. Ziet de computer een tweede diskdrive-interface, dan zal hij ervan uitgaan dat er maximaal vier diskdrives kunnen zijn aangesloten.
Bij het gebruik van één diskdrive weet de computer dat het oproepen van 'diskdrive B' niet mogelijk is. In dit geval zal dat geen foutmelding tot gevolg hebben, maar zal de computer u vragen 'diskette B' te plaatsen. Door deze ingebouwde 'intelligentie' lijkt het of u toch gebruik maakt van twee diskdrives. In het begin lijkt dit nogal verwarrend, maar omdat de computer zelf bijhoudt of diskette A of diskette B het laatste is gebruikt, kunnen hiermee in de praktijk weinig vergissingen ontstaan. Ook tijdens het kopiëren zal de computer de gebruiker er steeds op attenderen dat er van diskette dient te worden gewisseld tot de computer alle door u gewenste gegevens heeft gekopieerd.

Bij twee diskdrives zal het kopiëren natuurlijk veel sneller verlopen, omdat dit zonder het wisselen van diskettes gepaard gaat.
Heeft u twee diskdrives die niet met elkaar zijn verbonden, maar die elk op een eigen interface zijn aangesloten, dan bevinden die interfaces zich in verschillende sloten en heeft u de beschikking over de diskdrives A en C. Dus zodra de computer twee verschillende interfaces herkent, gaat hij ervan uit dat er minimaal twee en maximaal vier diskdrives zijn aangesloten.
Bij het oproepen van een andere diskdrive kijkt de computer eerst of de opgeroepen diskdrive aanwezig is. Zo ja, dan gaat de computer gewoon door; zo nee, dan zal de computer vragen om een andere diskette te plaatsen om op die manier de niet-aangesloten diskdrive te vervangen.
Daar we ervan uitgaan dat er voorlopig maar één diskdrive aanwezig is en de werking niet afwijkt, bespreken we alleen indien nodig de afwijkingen die zich voor kunnen doen bij het gebruik van méér dan één diskdrive.
Na het aanbrengen van de interface in één van de daarvoor bestemde sloten en het aansluiten van de diskdrive(s) kan men de spanning van de diskdrive inschakelen.
Zet altijd als laatste de computer aan zonder dat zich een diskette in de diskdrive bevindt. De computer zal nu opstarten in Disk BASIC. Alle extra commando's voor Disk BASIC bevinden zich in een ROM (leesgeheugen) waarmee de interface is uitgerust. Om direct van MSX-DOS gebruik te kunnen maken, heeft men een andere opstartprocedure nodig die in het volgende hoofdstuk ter sprake komt.
De opstartprocedure is anders omdat MSX-DOS zich niet in een ROM in de interface bevindt, maar op diskette staat en tijdens de opstartprocedure automatisch door de computer wordt ingelezen.

1.3 MSX-disketteformat

Naast de grote verscheidenheid in diskette-afmetingen die binnen de MSX-standaard is teruggebracht tot de 3 1/2 inch-diskette, kent MSX wel vier verschillende indelingen voor de diskette, hetgeen men dus een 'format' noemt. Hier komt dan ook de term 'formatteren' vandaan. Hoe men moet formatteren komt in hoofdstuk 4 en 5 ter sprake. We beperken ons in dit stadium alleen tot het eindresultaat van het formatteren.
Een diskette heeft in principe een opslagcapaciteit van ca. 500 Kbyte. Een nieuwe diskette kan, zoals we reeds weten, niet zonder meer worden gebruikt om informatie op te slaan.
De denkbeeldige cirkels die tijdens het formatteren worden aangebracht noemen we 'tracks' en die tracks zijn onderverdeeld in sectoren. Binnen deze indeling of format worden de





tracks genummerd van 0 tot 79 (bij 80 tracks) of van 0 tot 39 (bij 40 tracks). Ook de sectoren worden genummerd en wel van sector 1 tot sector 8 of 9, afhankelijk van het aantal sectoren die er per track worden ingedeeld.
MSX kent dan ook de volgende formats:

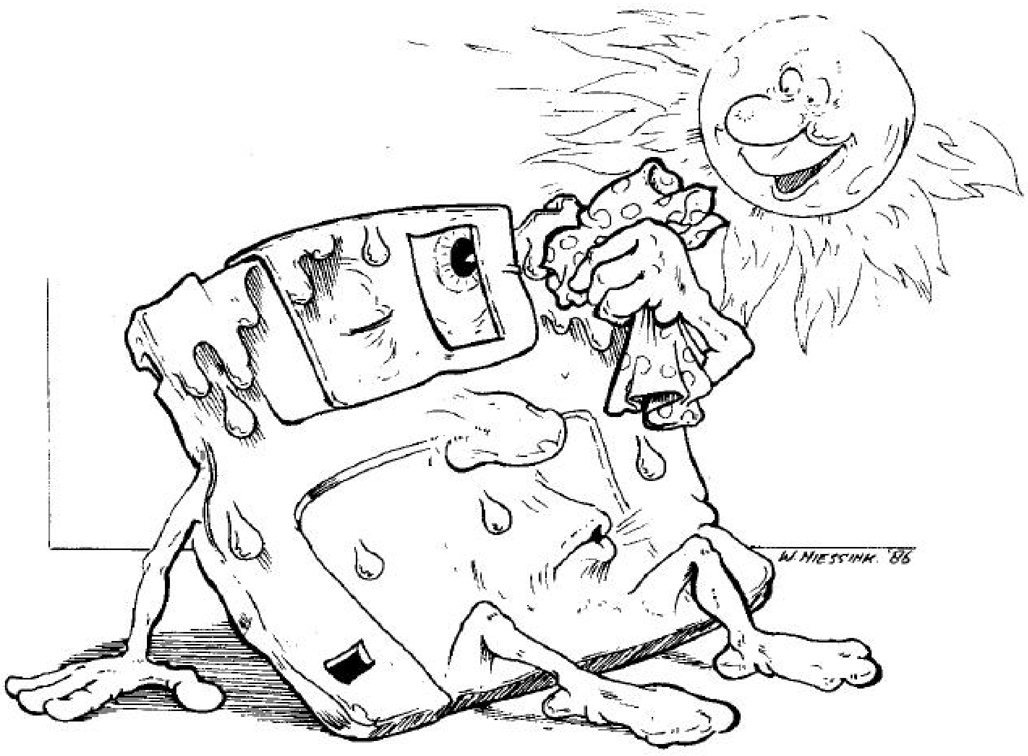
80 of 40 tracks
9 of 8 sectoren per track
enkel- of dubbelzijdig

In totaal zijn dit dus vier verschillende mogelijkheden als we enkel- en dubbelzijdig buiten beschouwing laten, daar dit laatste op het format geen invloed heeft. Bij het formatteren worden de gegevens omtrent het format van de diskette weggeschreven (opgeslagen) naar track 0, sector 1.
Totaal bevinden zich dus respectievelijk 720, 640, 360 of 320 sectoren op één diskette. Eén sector kan 512 bytes bevatten waardoor een diskette met 80 tracks en 9 sectoren per track, maximaal 368.640 bytes aan informatie kan opslaan.
Het format van de diskette wordt bepaald door de interface. In de interface bevindt zich een systeemvariabele die het format van de diskettes bevat.
Tijdens het formatteren reserveert het systeem ruimte voor een inhoudsopgave (de 'directory') en ruimte voor een file locatietabel (File Allocation Table - FAT). Hierdoor blijft er in eerste

instantie een effectieve opslagruimte over van 362.496 bytes. Bij het wegschrijven van een file of bestand naar de diskette wordt de filenaam opgeslagen in één sector, iedere filenaam en de daarbij behorende gegevens staan dus per file in één sector.
In de directory worden opgeslagen: de filenaam, de lengte van die file in bytes, het track- en sectornummer van de FAT die de opslaggegevens van de file bevat, de datum waarop de file werd gemaakt of voor het laatst werd bijgewerkt en, indien uw computer is uitgerust met een interne klok, wordt ook de tijd opgeslagen waarop de file voor het laatst naar de diskette werd weggeschreven.
In de FAT staat welke track(s) en welke sectoren zijn gebruikt om de file op te slaan en wel in een dusdanige vorm dat het begin van de file als eerste wordt opgeslagen (track- en sectornummer) met daarachter het track- en sectornummer van het vervolg van de file en ten slotte het einde van de file, ook weer het track- en sectornummer. Tevens wordt de nog beschikbare vrije ruimte van de diskette in de FAT vastgelegd. De directory en de FAT zorgen samen met Disk BASIC of MSX-DOS voor de juiste behandeling van uw files op de diskette. Vandaar dat het uiterst belangrijk is om diskettes te gebruiken die voldoen aan het juiste format, zodat de computer direct de juiste track en sector kan lezen van de directory. Aan de hand van de daar opgeslagen gegevens kan dan worden doorgegaan naar de FAT. Daar vindt de computer het track- en sectornummer van de gezochte file. De computer leest één voor één de sectoren in tot en met de laatste sector die in de FAT staat vermeld voor die bewuste file.
Ondanks de MSX-standaard zal in de praktijk blijken dat niet alle diskettes uitwisselbaar zijn. Alleen diskettes die zijn geformatteerd volgens het format dat in uw interface is opgeslagen, kunt u zonder problemen gebruiken.

1.4 Bewaren van diskettes

De 3 1/2 inch-diskettes vragen niet om speciale aandacht wat het bewaren betreft. Toch geven we een aantal tips die van pas kunnen komen om uw diskettes te beschermen tegen schade.
Eén van de belangrijkste punten is dat we ons goed moeten realiseren dat de informatie langs magnetische weg wordt opgeslagen. Ook het wissen van die informatie gebeurt langs magnetische weg en helaas bij onzorgvuldige behandeling niet alleen door de diskdrive. In uw huis bevinden zich meer magnetische velden dan u op het eerste gezicht zou vermoeden!
In luidsprekers bijvoorbeeld, bevinden zich over het algemeen zeer sterke magneten. Een diskette die u even achteloos op zo'n luidspreker of luidsprekerbox legt, kan door het magnetische veld dat de luidspreker uitstraalt zijn informatie verliezen.
Ook televisietoestellen vertonen over het algemeen een sterke



magnetische straling, die voldoende kan zijn om uw kostbare informatie op de diskette te vernietigen. Denk eraan dat ook uw monitor een soort televisietoestel is dat magnetische velden uitstraalt.
De voeding van uw computer kan ook informatie op een diskette verminken. Dit geldt trouwens voor alle voedingen: de adapter of voeding van een cassetterecorder, de voeding van de printer enz.
Enige voorzichtigheid is dus toch op zijn plaats!
Het beste kunt u de diskettes in speciaal daarvoor gefabriceerde dozen opbergen. Deze zijn stevig en in sommige gevallen zelfs voorzien van een slot om uw privé-informatie extra te beveiligen. Tevens zijn de diskettes dan goed beschermd tegen stof en vuil, de grootste vijand van een diskette.
Beseft u wel dat 80 sporen naast elkaar op een kleine schijf een breedte hebben van minder dan een kwart millimeter? Eén stofdeeltje, hoe klein ook, is bijna net zo groot! Als dat stofdeeltje per ongeluk tussen de kop en de diskette komt, kan dat lelijke krassen veroorzaken. Daarbij kan het magnetische oppervlak zodanig worden beschadigd dat de informatie niet meer foutloos wordt gelezen of weggeschreven.
Ten slotte geven we nog een aantal tips die misschien nog belangrijker zijn dan de hiervoor genoemde aanwijzingen.
Het is van belang de diskettes niet aan direct zonlicht bloot te stellen, niet te vouwen en geen paperclips in de buurt of op een diskette te laten slingeren. Laat een diskette ook niet langer dan nodig is in de diskdrive zitten. Neerslag en as van een sigaret e.d.

kunnen ook schade veroorzaken. Een zorgvuldige behandeling staat voorop bij het werken met diskettes.

Maar toch..., ook bij perfecte behandeling van de diskettes kan het voorkomen dat er informatie verloren gaat. U kunt per ongeluk zelf informatie wissen, diskdrives kunnen een fout maken en ten slotte slijten de diskettes op den duur toch ook. Werk daarom altijd met een diskette die een kopie bevat van uw originele programma. Gebruik die kopiediskette (bij computers 'back-up' genaamd) dus als werkdiskette! Mocht het gebeuren dat uw back-up diskette moeilijkheden geeft, kopieer dan de originele diskette naar een andere 'nieuwe' diskette, maar gebruik de originele diskette nooit als werkdiskette, maar alleen als informatiebron of programmadrager.
Belangrijk is het kopiëren na het bijwerken van gegevens. Maak er een gewoonte van om bij langdurige en ingrijpende wijzigingen in programma's of informatie regelmatig een tweede werkkopie te maken! Natuurlijk niet op dezelfde diskette, maar altijd op een andere, speciaal daarvoor gereserveerde diskette. Staat de kopie per ongeluk toch op de eerste werkdiskette, dan kunt u daar nooit bij komen als die diskette eventueel is beschadigd en lezen vanaf die diskette niet meer mogelijk is. Bewaar de originele en tweede back-up diskette altijd apart van de werkdiskettes en voorzie ze van etiketten met daarop de belangrijkste gegevens en de laatste back-up datum. Schrijf niet rechtstreeks op de hoezen van de 5 1/4 inch-diskettes, dat kan krassen veroorzaken in het magnetische materiaal.
Van belangrijke programma's is het verstandig om een extra back-up diskette te maken en van zelfgemaakte programma's is een listing op papier beslist geen overbodige luxe.

HOOFDSTUK 2

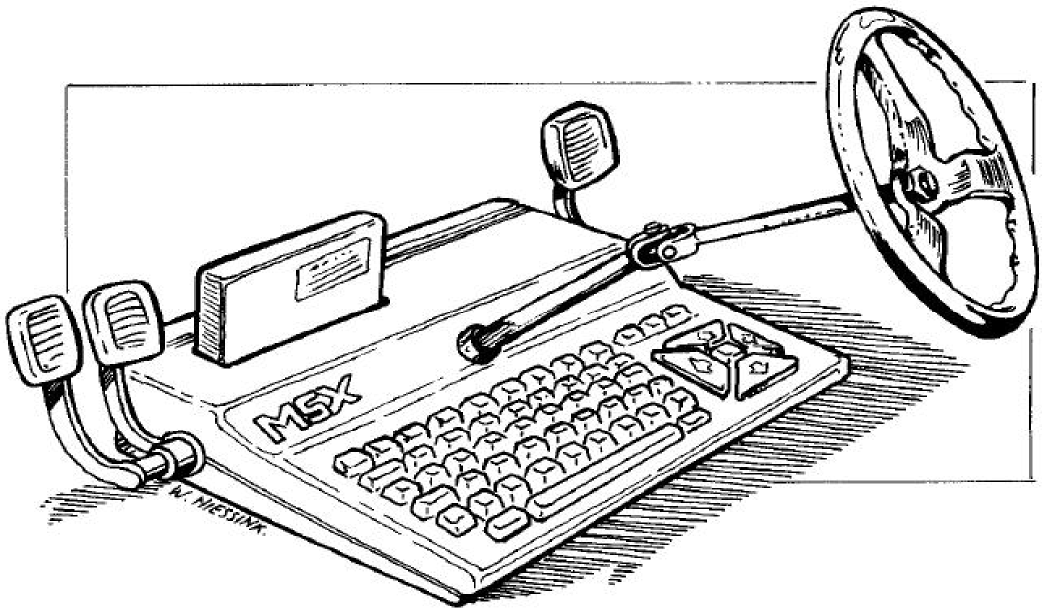
WAT IS EEN BESTURINGS-SYSTEEM?

2.1 Het besturingssysteem

Een besturingssysteem is 'de link' tussen u als gebruiker en de computer. Het verzorgt de communicatie tussen u, de hardware en de software. U kunt via een besturingssysteem rechtstreeks met de computer communiceren, met de printer, de diskdrives en met eventueel andere aangesloten apparatuur.
Een besturingssysteem zorgt ervoor dat u door middel van relatief eenvoudige commando's diverse apparaten kunt laten werken, zonder dat u zich hoeft te bekommeren om de mechanische of elektrische werking van die apparatuur. U kunt het vergelijken met de elektriciteitsvoorziening in huis, als u bijvoorbeeld het commando 'licht aan' wilt geven, zet u de lichtschakelaar op 'aan' en zal naar alle waarschijnlijkheid het licht gaan branden.



Ook hier behoeft u zich niet te bekommeren om de exacte elektrische of mechanische werking van de lichtketen.
Het besturingssysteem is een stuk systeemsoftware dat speciaal is ontworpen voor uw type computer. U kunt MSX BASIC ook zien als een besturingssysteem (programmeertaal genoemd), dat het mogelijk maakt de aangesloten apparatuur via BASIC aan te sturen, bijvoorbeeld de cassetterecorder voor het opnemen van programma's. De printer is ook te besturen met behulp van MSX BASIC en natuurlijk de diskdrive.
Het BASIC-besturingssysteem bevat een soort woordenboek dat is opgeslagen in de computer, meestal in ROM (het zgn. 'leesgeheugen' van de computer). Deze woordenlijst bevat alle woorden (commando's) die de computer moet herkennen. Sommige woorden kunnen alleen worden gebruikt in combinatie met een ander woord. De woordenlijst die bijvoorbeeld alle BASIC-commando's bevat noemt men met een mooi woord de BASIC-interpreter. Deze interpreter vertaalt het BASIC-commando in speciale instructies die de microprocessor kan uitvoeren. Na het invoeren van een commando zoekt de computer in de 'verklarende' woordenlijst naar de betekenis en zet het dan om naar de instructies die hij daar vindt. Kan de computer het commando niet vinden, dan zal er een foutmelding op het beeldscherm verschijnen en kunt u het met een ander commando nog eens proberen.
Heeft men een diskdrive aangesloten op de computer, dan zijn er een aantal extra commando's nodig voor het besturen van en het werken met de diskdrive. Bij de meeste computers moet men een apart disk-besturingssysteem aanschaffen om met de diskdrive te kunnen werken. MSX-computers vormen hierop gedeeltelijk een uitzondering, omdat de diskdrive wordt geleverd met een 'aanpassingsmoduul' (interface) die is voorzien van een ROM, die de benodigde extra commando's bevat.
Men kan dan werken met het zogenoemde MSX Disk BASIC. Daarnaast biedt MSX nog een tweede mogelijkheid om met de diskdrive te werken. Dit tweede systeem komt veel overeen met de systemen die bij duurdere Personal Computers worden gebruikt en heet Disc Operating System (kortweg DOS genaamd). Natuurlijk zal MSX-DOS alleen op een MSX-computer werken en niet op andere computers, tenzij men een groot deel van de instructies gaat veranderen.
MSX-DOS is een besturingssysteem dat het mogelijk maakt om met behulp van de DOS-commando's gegevens direct naar een diskette weg te schrijven; die gegevens noemt men dan een file of bestand. Men kan met MSX-DOS die weggeschreven files laten uitvoeren, randapparatuur aansturen (bijvoorbeeld printers en andere diskdrives), kopiëren e.d.
De woordenlijst van een Disk Operating System bevindt zich niet in ROM, maar staat altijd op een diskette (ook wel systeem-



diskette genoemd) en moet voor het gebruik in de computer worden opgeslagen (geladen). Daartoe zal de computer een gedeelte van zijn geheugen moeten reserveren. Deze ruimte noemt men de 'commando-interpreter' of 'commandoprocessor'. De commandoprocessor zal niet direct overgaan tot een foutmelding als het opgegeven commando niet voorkomt in de 'woordenlijst'. Als het commando niet voorkomt, zal de commandoprocessor namelijk eerst op de diskette kijken of het opgegeven 'woord' in de 'inhoudsopgave' (in computertermen: 'directory') van de diskette staat. Zo ja, dan wordt er gekeken of het een commandofile is waarmee de commandoprocessor kan werken. Na het uitvoeren van die commandofile is de commandoprocessor weer beschikbaar voor een volgend commando.
Als de commandoprocessor het 'woord' niet als commandofile in de 'directory' kan vinden zal hij een overeenkomstige foutmelding genereren en op het beeldscherm weergeven. De commandoprocessor heeft dan zijn taak voorlopig volbracht en wacht op een volgende instructie.
Verlaat men DOS, dan zal die gereserveerde ruimte weer vrijkomen voor andere functies. Om terug te keren naar DOS, dient dit eerst te worden geladen: of met een speciaal commando of door opnieuw opstarten van de computer.
Net als andere besturingssystemen zal ook MSX-DOS voor de gebruiker steeds belangrijker worden, omdat er een groot aantal softwareprogramma's worden geleverd die alleen werken onder MSX-DOS. De belangrijkste softwareprogramma's zijn o.a. PASCAL, een BASIC Compiler, FORTRAN, LOGO, Multiplan, Aackotext.

2.2 Opstarten en werken met MSX Disk BASIC

Na het aansluiten van de interface en het aanzetten van de randapparatuur kan men ten slotte de computer aanzetten. Op het beeldscherm verschijnt eerst een copyright mededeling:

```
MSX system
version 1.0      (of hoger)
Copyright 1983 by Microsoft
```

Daarna verschijnt op het beeldscherm de volgende vraag:

```
Enter date (M-D-Y):■
```

De datumaanduiding kan voor de verschillende landen anders zijn, zo zal de Amerikaanse datumaanduiding Y-M-D zijn.

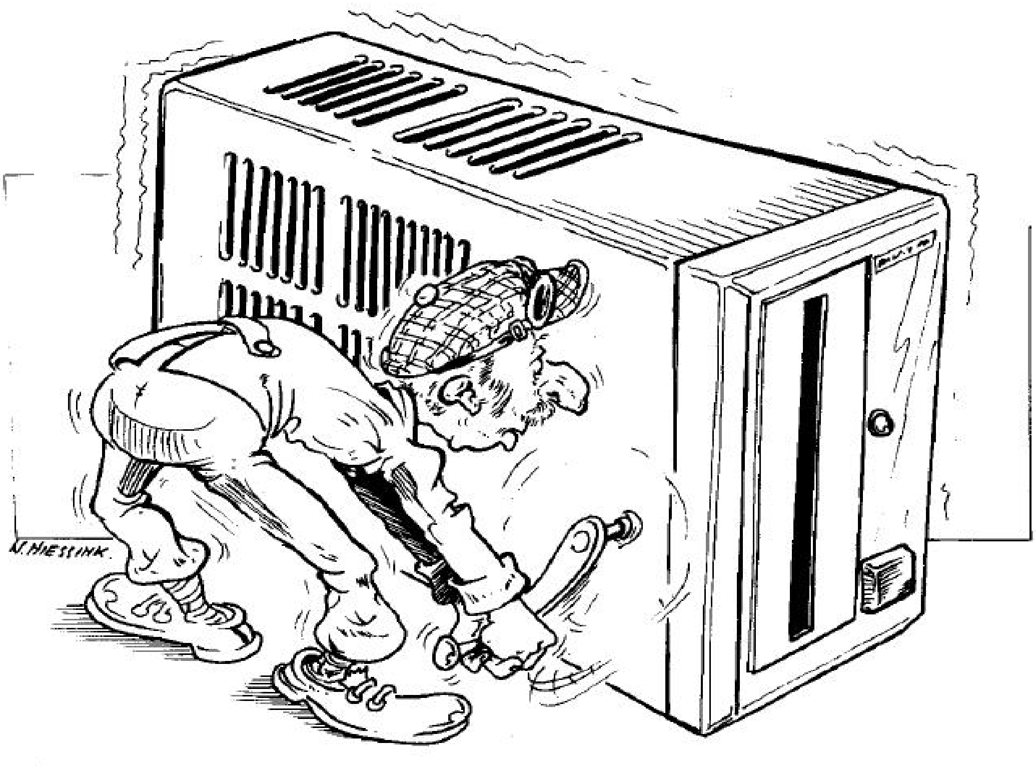
Na het invoeren van de juiste datum en een RETURN (of alleen RETURN) krijgt u de volgende boodschap:

```
MSX BASIC version 1.0      (of hoger)
Copyright 1983 by Microsoft
xxxxx Bytes free
Disk BASIC version 1.0     (of hoger)
Ok
■
```

Op de plaats van 'xxxxx' staat een getal dat afhankelijk is van uw computer, dit geeft de beschikbare vrije geheugenruimte voor programma's aan.
MSX Disk BASIC is nu beschikbaar voor gebruik. Het Disk BASIC bevat een aantal commando's die rechtstreeks de diskdrive activeren, de overige commando's worden gebruikt in combinatie met MSX BASIC-programma's om tijdens het programmaverloop informatie naar de diskette weg te schrijven of te lezen.
MSX Disk BASIC moet men dan ook zien als een 'hulptaal' ter aanvulling van het gewone MSX BASIC.
Daarentegen is MSX-DOS een 'commandotaal' die naast de commando's voor de diskdrive ook systeemcommando's bevat voor het werken met de andere aangesloten apparatuur.

2.3 Opstarten en werken met MSX-DOS

Het MSX-DOS besturingssysteem bevat een aantal specifieke commando's die niet mogelijk zijn met MSX Disk BASIC, doch sommige zijn er wél aan verwant. Om verwarring te voorkomen, zijn in appendix C de verschillen op een rijtje gezet. Het 'disk



operating system' moet dus altijd vanaf een diskette worden gelezen en worden opgeslagen in de commandoprocessor. Hebben we een computer die niet voldoende ruimte heeft om een deel van het geheugen te reserveren voor de commandoprocessor, dan kan MSX-DOS niet worden geladen. MSX-DOS werkt dan ook alleen op MSX-computers met een geheugencapaciteit van 64 K RAM of meer (RAM wordt ook wel 'werkgeheugen' genoemd).

Het laden van het besturingssysteem kan op twee manieren geschieden: met de 'koude' of de 'warme' start. Indien het besturingssysteem 'koud' wordt opgestart wil dat zeggen: alle apparatuur staat uit, eerst moet alle randapparatuur worden aangezet en ten slotte de computer; onmiddellijk daarna moet de MSX-DOS-systeemdiskette in diskdrive A worden geplaatst. Staat de computer al aan (in Disk BASIC), dan kunnen we ook de 'warme' start toepassen. Plaats daartoe eerst de MSX-DOS-systeemdiskette in diskdrive A en druk dan op de reset-knop van de computer. Als uw computer geen reset-knop heeft, kunt u MSX-DOS alleen maar met de 'koude start' laden.

De computer herkent de interface van de diskdrive en zal opstarten in Disk BASIC, hiertoe zal de computer de eerste sector van track nul lezen en de daarin vastgelegde gegevens opslaan in het geheugen. Tevens komt de copyright boodschap op het beeldscherm die we in de vorige paragraaf hebben gezien. Nu zal het systeem gaan zoeken naar een 'systeemfile' die ervoor moet zorgen dat er geheugenruimte wordt gereserveerd voor de commandoprocessor. Vindt de computer geen 'systeemfile'

dan blijft hij in MSX Disk BASIC en volgt het systeem de procedure die we in de vorige paragraaf hebben beschreven.
Wordt de systeemfile wél gevonden, dan zullen alle benodigde parameters worden geïnitialiseerd en de adressen voor de commandoprocessor gereserveerd. In de systeemfile staat het versienummer van het disk operating system dat tevens op het beeldscherm wordt weergegeven.

```
MSX-DOS version 1.07          (of hoger)
Copyright 1984 by Microsoft
```

Hierna wordt de op diskette aanwezige commandofile gelezen en alle commando's opgeslagen in de commandoprocessor. Na het inlezen volgt weer een mededeling omtrent het versienummer van de commandofile:

```
COMMAND version 1.07          (of hoger)
```

Ten slotte zoekt de computer op de diskette naar een speciale file die ervoor zorgt dat er automatisch een programma kan worden geladen. Zo'n file is door de gebruiker zelf te creëren en draagt de naam 'AUTOEXEC.BAT-file', maar daarover meer in hoofdstuk 8.
Als er geen AUTOEXEC.BAT-file aanwezig is op de diskette verschijnt de volgende vraag:

```
Current date is Sun 1-01-1984
Enter new date: ■
```

Na het invoeren van de datum en RETURN (of alleen RETURN) verschijnt de MSX-DOS-cursor ten teken dat het systeem (lees: computer) gereed is om opdrachten van de gebruiker te ontvangen en indien mogelijk uit te voeren.

2.4 Het invoeren van de datum

Na het opstarten vraagt de computer om de datum, zowel bij Disk BASIC als DOS. Deze datum wordt bij uw programma's, bestanden of files opgeslagen zodra u deze wegschrijft naar de diskette. Hierdoor kunt u direct zien welke versie van één van uw eigen programma's de laatste is.
De vraag om de datum bij Disk BASIC luidt:

```
Enter date (M-D-Y):■
```

Bij MSX-DOS zijn de systeemfiles al voorzien van een datum die de computer leest en als 'voorlopige' huidige datum aanneemt. De vraag om de datum bij MSX-DOS luidt dan ook:



```
Current date is Sun 1-01-1984
Enter new date: ■
```

In Disk BASIC wordt de juiste invoer-volgorde al aangegeven terwijl dat bij MSX-DOS ontbreekt. Voor beide geldt als juiste volgorde:

M-D-Y

De datumaanduiding kan voor de verschillende landen anders zijn, zo zal de Amerikaanse datumaanduiding Y-M-D zijn.

Hierin is:

M de maand; 1 of 2 cijfers van 1 tot 12
D de dag; 1 of 2 cijfers van 1 tot 31
Y het jaar; 2 cijfers van 00-99 of 4 cijfers van 1980-2099

Bij het invoeren van twee cijfers voor het jaartal, krijgt u met de getallen 00-79 de jaartallen van 2000-2079 en met de getallen 80-99 de jaartallen 1980-1999. Maar u kunt zelfs experimenteren tot het jaar 2099 als u het jaartal met vier cijfers invoert (2080-2099). Het komt er dus op neer dat alleen de jaartallen vanaf 1980 tot en met 2099 mogelijk zijn.
Iedere datum wordt geaccepteerd als die voldoet aan de hiervoor genoemde regels en als het een bestaande datum is. Zo zal bijvoorbeeld 29 februari 1991 een foutmelding ten gevolge hebben daar 1991 geen schrikkeljaar is en 29 februari dus ook niet bestaat in dat jaar.

Als scheidingstekens tussen de maand, dag en het jaar kunnen de '-', '/' en de '.' worden gebruikt.
De computer berekent aan de hand van de ingevoerde datum welke dag het is. Daar de programmeertaal Engels is, zal de dag ook in het Engels worden weergegeven.
Bijvoorbeeld:

4-18-85 wordt Thu 18-4-1985
6/20/72 wordt Mon 20-6-2072
11.28.1990 wordt Thu 28-11-1990

Maakt u tijdens het invoeren van de datum een fout, dan kunt u met de backspace-toets (BS) de reeds ingevoerde karakters verwijderen en de juiste datum alsnog invoeren. Bent u de datum nog aan het invoeren, maar u wilt er liever van afzien, dan kunt u met <CONTROL>-C de tot nu toe ingevoerde gegevens tenietdoen en de computer gaat door met het opstarten.
Het intoetsen van een <RETURN> achter de vraag naar de datum zonder een datum in te voeren, laat bij MSX-DOS de datum zoals hij was (de 'current date'). Bij MSX Disk BASIC zal er dan geen datum aanwezig zijn in de computer.
Tijdens het werken met de files kunt u de datum onder MSX-DOS, indien nodig, alsnog opgeven of veranderen. Meer hierover vindt u in het hoofdstuk over de specifieke MSX-DOS-commando's.

2.5 MSX-DOS-systeemfiles

Uw MSX-DOS-systeemdiskette bevat minimaal twee files, namelijk de files met de naam COMMAND.COM en MSXDOS.SYS. U kunt in het volgende hoofdstuk lezen dat de files eindigend op .COM commandofiles in machinetaal zijn en commando's bevatten die worden opgeslagen in de commandoprocessor. De MSXDOS.SYS-file zorgt voor het initialiseren van de computer en het laden van de commando's die zijn opgeslagen in de COMMAND.COM-file. Minimaal staan er dus twee files op de systeemdiskette, doch het is mogelijk dat de diskette meer files bevat. Het is verstandig om direct na aanschaf en opstarten van MSX-DOS een kopie te maken van de systeemdiskette.
Raadpleeg eerst hoofdstuk 6 om een nieuwe diskette gebruiksklaar te maken voor uw computer om daarna alle files te kopiëren, zodat u een eerste 'back-up' heeft van de systeemdiskette om mee te experimenteren en te werken.

Uw 'back-up'-diskette kan daarna allerlei soorten gegevens bevatten: normale programma's, machinetaalprogramma's, ASCII-files, tekstfiles enz. Al deze files dienen voor u als gebrui-

ker herkenbaar te zijn, vandaar dat een filenaam kan worden voorzien van een toevoegsel (de 'extension' genaamd), waardoor u direct kunt zien met wat voor een file u te maken heeft. De verschillende wel- en niet-toegestane extensions (toevoegingen) komen in het volgende hoofdstuk aan de orde. Het is nu alleen van belang dat u weet dat de file COMMAND het achtervoegsel .COM van commandofile heeft en de file MSXDOS het achtervoegsel .SYS dat op het systeem zelf duidt.

HOOFDSTUK 3
FILES EN BESTANDEN

3.1 Wat zijn files?

Een file bestaat uit een verzameling gegevens die bij elkaar behoren. Zo kan een file op uw diskette bijvoorbeeld de namen en adressen van uw vrienden of van leden van een vereniging bevatten, hetgeen men een (adressen-) bestand noemt. Dit bestand geeft u dan natuurlijk de naam 'kennis' of 'vrienden'. Men dient altijd een naam te kiezen die de inhoud van het bestand enigszins aangeeft. Het zou onlogisch zijn het bestand dat de namen van uw vrienden of de leden van een vereniging bevat de naam 'FIETS' te geven.

Steeds als u iets wegschrijft naar de diskette wordt er een bestand of file gecreëerd. Voordat u een bestand of file naar een diskette kunt wegschrijven moet u dat eerst kenbaar maken aan het systeem. Bij Disk BASIC noemen we dat het 'openen' van een bestand en bij 'DOS' kopiëren we een file vanaf het toetsenbord (console) naar de diskette. In beide gevallen moet dat altijd gebeuren onder vermelding van een naam waaronder het bestand of de file kan worden teruggezocht. Deze naam wordt opgenomen in de inhoudsopgave van de diskette met de noodzakelijke gegevens om de file later weer te kunnen inlezen. Een naam moet minstens uit één en mag maximaal uit acht tekens of karakters bestaan. De toevoeging (extension) is niet noodzakelijk, maar kan handig zijn bij het onderscheiden van de verschillende soorten opgeslagen gegevens en mag dan uit maximaal drie karakters bestaan. Deze mogen zowel letters als cijfers bevatten. Gebruikt men niet alle acht posities voor de naam, dan zal het systeem de naam aanvullen met spaties tot acht posities en bij de extension tot drie posities.

In dit boek wordt de benaming bestand gebruikt voor bij elkaar behorende gegevens. De benaming programma wordt gebruikt voor het aangeven van opeenvolgende bewerkingen of uitvoering van bij elkaar behorende of samengevoegde commando's.

De benaming file is de 'Engelse' verzamelnaam voor bestanden en programma's, deze term wordt gebruikt in die gevallen waar de benaming 'bestand' of 'programma' niet geheel op zijn plaats is, bijvoorbeeld om een 'tekstfile' aan te geven.

3.2 Filenamen en filenaam extensions

De complete naam kan dus uit twee delen bestaan, de eigen naam en de 'extension' die de soort aangeeft, beide moeten altijd van elkaar zijn gescheiden door een punt. De extension kan, zoals we hebben gezien, de soort file aangeven. De toevoeging .COM die we in het vorige hoofdstuk hebben leren kennen is speciaal bedoeld voor een commandofile.
Een aantal extensions hebben voor de commandoprocessor dan ook een speciale betekenis en mogen alleen worden gebruikt in die gevallen waar de commandoprocessor met de file kan en moet werken.
Nemen we als voorbeeld: MIJNPROG.BAS, dan is MIJNPROG de naam van de file en BAS de extension, die aangeeft dat het om een BASIC-programmafile gaat. De naam en de extension mogen zowel in hoofdletters als kleine letters worden opgegeven, MSX-DOS en Disk BASIC zullen bij het gebruik van kleine letters deze eerst lezen en daarna omzetten naar hoofdletters.
In principe is men vrij in het bepalen van de extension, maar gezien de MSX-standaard is het verstandig de volgende richtlijnen aan te houden voor de extension.

Extension	*Omschrijving*
ASC	Een ASCII-file.
ASM	Een assembler-programmafile.
BAK	Een BAcK-up van een willekeurig bestand of programma.
BAS	Een BASIC-programma.
BAT	Een BATCH-programma t.b.v. het besturingssysteem.
BFR	Een tekstfile, automatische toevoeging bij het tekstverwerkende programma Aackotext.
CBL	Een COBOL-programma.
COM	Een commandofile t.b.v. het besturingssysteem.
HEX	Een hexadecimale (machinetaal) file.
INT	Een intermediate (tijdelijke) file.
MAC	Een MACRO assembler-programma.
PIC	Een PICture (afbeelding) datafile.
PL1	Een PL/1 programma.
REL	Een RELocatable modulaire file.
SYM	Een SID symboolfile.

SYS	Een systeemprogramma t.b.v. het besturingssysteem.
TEX	Een tekstfile, zie ook .TXT.
TXT	Een tekstfile, mogelijk een automatische toevoeging bij een tekstverwerkend programma.

Gebruikt u andere namen als extension, dan zullen er geen problemen ontstaan met andere programma's of nieuwere versies van het besturingssysteem. Ofschoon u in bovenstaande opsomming heeft kunnen zien dat er extensions bestaan die mogelijk door bepaalde programma's zelf worden gegenereerd, is het mogelijk om met één commando naderhand ongewenste extensions te veranderen. Naast de vastgestelde betekenis van de extension, heeft MSX ook een aantal filenamen die een speciale betekenis hebben. Men mag deze benamingen dan ook niet gebruiken voor eigen files of bestanden.
De eerste drie filenamen zijn gereserveerde namen onder MSX-DOS en de laatste onder Disk BASIC.

MSXDOS.SYS	Deze file maakt deel uit van het Disk Operating System van MSX-DOS.
COMMAND.COM	Deze file bevat de commando's voor MSX-DOS.
AUTOEXEC.BAT	Deze file bevat commando's voor het automatisch opstarten van programma's onder MSX-DOS.
AUTOEXEC.BAS	Deze file bevat commando's om een BASIC-programma automatisch te laten RUNnen bij het aanzetten van de computer indien er geen MSX-DOS aanwezig is.

Het is mogelijk dat er op de MSX-DOS-systeemdiskette meer files staan. Schaft u bijvoorbeeld de MSX-DOS-systeemdiskette van Philips aan, dan treft u ook de file DOSHLP.COM aan; een gebruikershulp om het systeem te leren kennen. Alle extra namen die op de MSX-DOS-systeemdiskette staan moet u beschouwen als gereserveerde namen. Gebruik deze dan ook niet voor eigen files, bestanden of programma's.

MSX kent een aantal benamingen die speciaal zijn bestemd voor de aangesloten apparatuur. Deze namen gelden zowel voor Disk BASIC als voor MSX-DOS en vallen onder de gereserveerde namen.

AUX	Een naam die refereert naar (de input of output van) een willekeurig niet-gespecificeerd extern aangesloten apparaat.
CON	Dit is de benaming voor het toetsenbord (input) en het beeldscherm (output).
LST	Dit is de benaming voor de printer (LiST apparaat).
PRN	Eveneens een benaming voor de PRiNter.
NUL	Deze benaming dient als naam zodra het systeem naar een filenaam vraagt terwijl u geen file wilt creëren.

Ook bij het toevoegen van een extension aan één van de naar de randapparatuur refererende namen met de bedoeling om deze als filenaam te kunnen gebruiken, zal het systeem het eerste deel (de eigenlijke naam van de file) herkennen als referentie naar een aangesloten apparaat en niet als een filenaam. Zo zal bijvoorbeeld de benaming A:LST.TXT door het systeem worden gezien als een referentie naar de printer en niet naar een bestaande tekstfile.
Uw filenaam mag bestaan uit alle letters van het alfabet en uit de cijfers 0 tot en met 9, tevens zijn er nog een aantal andere karakters toegestaan. De volgende karakters zijn toegestaan voor zowel de naam als de extension.

De letters A - Z
De cijfers 0 - 9
$ & # % ‘ ( ) -
@ \ ^ { } ñ ’ !

Voorbeelden:

MYNPROG.BAS	BASIC programma met de naam ‘mynprog’
BRIEF1.TXT	Tekstfile met de naam ‘brief1’
MULTIPL.HEX	Machinetaalprogramma met de naam ‘multipl’

3.3 Wildkarakters

Er zijn twee karakters die beslist niet mogen worden gebruikt in de namen van files, bestanden of programma’s, dit zijn de zogenoemde ‘wildkarakters’. Deze wildkarakters (‘wild cards’) kunnen we toepassen bij het zoeken naar bepaalde files, bestanden of programma’s waarvan we de juiste benaming niet meer weten. Deze speciale karakters zijn: de asterisk (*) en het vraagteken (?).
Het vraagteken (?) in een naam of extension vervangt één willekeurig karakter op die plaats. De naam

```
TEST?RUN.COM
```

bestaat dus uit 8 karakters en begint met TEST en eindigt op RUN en heeft de extension .COM. Op de plaats van het vraagteken staat dus één van de hiervoor genoemde (toegestane) karakters. Roepen we bijvoorbeeld de inhoudsopgave van een diskette op, dan kunnen we de volgende namen op het beeldscherm verwachten als ze aanwezig zijn op die diskette:

```
TEST1RUN.COM
TEST2RUN.COM
TESTARUN.COM
```

Het gebruik van meer vraagtekens is natuurlijk ook mogelijk, nemen we bijvoorbeeld:

```
?EST?RUN.COM
```

dan zullen de volgende namen hieraan voldoen en op het beeldscherm verschijnen:

```
BEST1RUN.COM
TEST2RUN.COM
RESTARUN.COM
1EST9RUN.COM
NESTZRUN.COM
```

Het vraagteken kan één karakter vervangen, meer vraagtekens vervangen dus ook meer karakters. Hierdoor is het mogelijk om de gehele naam of extension te vervangen door vraagtekens. De volgende naam zal dus overeenkomen met alle namen die gelijk zijn aan de opgegeven naam zonder naar de extension te kijken:

```
TESTPROG.???
```

De volgende namen zullen hieraan voldoen:

```
TESTPROG.COM
TESTPROG.BAS
TESTPROG.TXT
TESTPROG.ASC
```

Maar ook:

```
TESTPROG.AA
```

zal hieraan voldoen, omdat deze uit twee letters bestaande extension in de inhoudsopgave van de diskette werd aangevuld

met een spatie en het achterste vraagteken die spatie als een willekeurig karakter herkent.

De asterisk (*) kan meer dan één karakter vervangen, waardoor het voorgaande voorbeeld met de drie vraagtekens op de plaats van de extension kan worden vervangen door:

```
TESTPROG.*
```

Dit resulteert in hetzelfde effect en de volgende namen komen dan ook weer op het beeldscherm:

```
TESTPROG.COM
TESTPROG.BAS
TESTPROG.TXT
TESTPROG.ASC
TESTPROG.AA
```

Zoals u ziet kan de asterisk inderdaad meer vraagtekens achter elkaar vervangen. De asterisk vervangt tevens een willekeurig aantal karakters. Of de naam nu uit één of uit vijf karakters bestaat, de asterisk vervangt ze allemaal. Bij het opgeven van vijf vraagtekens kijkt het systeem alleen naar namen die uit vijf of minder karakters bestaan. Met de asterisk zal zowel een naam van één, twee tot en met acht karakters voldoen. Dit is erg gemakkelijk als u bijvoorbeeld alle files met dezelfde naam maar met een andere extension op het beeldscherm wilt zien. Het toe-



passen van een asterisk voor zowel de naam als de extension zal dus naar alle files op de diskette refereren.

```
*.*
```

De asterisk kunt u op die manier dus ook gebruiken bij het zoeken naar tekstfiles waarvan u de juiste benamingen niet meer weet. Het volgende voorbeeld refereert dus naar alle tekstfiles ongeacht hun eigen naam; achter alle tekstfiles dient natuurlijk wel de extension .TXT te zijn opgenomen:

```
*.TXT
```

De volgende tekstfiles zullen worden weergegeven op het beeldscherm:

```
BRIEF1.TXT
TESTPROG.TXT
BRIEFA.TXT
LIJST-A.TXT
```

Hoewel de verschillende commando's voor Disk BASIC en MSX-DOS nog niet zijn besproken, is hier een waarschuwing op zijn plaats! De commando's

```
DEL *.*
```

of

```
ERASE *.*
```

zullen bij MSX-DOS alle gegevens van uw diskette wissen en alle informatie en de data gaan dus verloren.

Bij MSX Disk BASIC zal het commando

```
KILL "*.*"
```

alle gegevens wissen. Gebruik daarom nooit wildkarakters bij de commando's DEL, ERASE of KILL. Meer over deze commando's vindt u in de hoofdstukken 4 en 7, evenals de toepassing van de wildkarakters bij de verschillende commando's.

HOOFDSTUK 4 MSX DISK BASIC

4.1 Wat is MSX Disk BASIC?

MSX Disk BASIC is een uitbreiding van het gewone MSX BASIC. Met MSX Disk BASIC kan informatie zowel vanaf een diskette worden gelezen als er naartoe worden weggeschreven. Met specifieke Disk BASIC-commando's is het ook mogelijk om gedeelten 'uit' een file te lezen.
Normaal wordt een compleet programma, bestand of file vanaf de diskette gelezen en in de computer opgeslagen. Bevat een bestand de gegevens achter elkaar, dan spreken we van een sequentieel bestand. We dienen de volgorde waarin de gegevens zijn weggeschreven te kennen om ze weer te kunnen lezen. Het lezen en schrijven van en naar de diskette gaat net zo in zijn werk als bij het gewone MSX BASIC, alleen dient men voor de naam van het bestand of programma niet de benaming van de cassetterecorder op te nemen, maar de benaming van de diskdrive, bijvoorbeeld 'A:DATA' of 'B:PROGA.BAS'. Bij het werken met sequentiële bestanden kent Disk BASIC het commando 'APPEND' om een reeds bestaand sequentieel bestand uit te breiden. Dit commando werkt niet bij een cassetterecorder.
Zodra men de beschikking heeft over een diskdrive wordt het maken van en werken met 'Random Access Files' interessant. Men dient bij een sequentieel bestand alle gegevens in één keer in te lezen om bepaalde data te vinden, bij een random access bestand kunnen we rechtstreeks naar bepaalde data vragen, al staat die halverwege het bestand. Na de behandeling van de specifieke Disk BASIC-commando's, zullen we in het volgende hoofdstuk de eigenschappen en het werken met de Random Access File uitgebreid bespreken.

4.2 Het opstarten

Het opstarten van de computer is al besproken in paragraaf 2.2.

We noemen de verschillende opstartprocedures om in MSX Disk BASIC te komen nog even in het kort.
Indien men de computer aanzet zonder dat er een diskette is geplaatst, zal MSX Disk BASIC direct beschikbaar zijn na het invoeren van de datum (zie paragraaf 2.4).
Start men op met een diskette in diskdrive A, dan mag deze diskette niet de voor MSX-DOS benodigde systeemfiles bevatten (zie paragraaf 2.5).
Bevat de in diskdrive A geplaatste diskette toch de beide MSX-DOS-systeemfiles, dan dient men, nadat MSX-DOS is geactiveerd, met het commando 'BASIC' over te gaan naar Disk BASIC (zie paragraaf 7.4).

Na het opstarten is het belangrijk om te kunnen zien wat er op de diskette staat die we in de diskdrive plaatsen. Hiertoe kent Disk BASIC het commando 'FILES':

```
Ok
FILES
File not found
Ok
■
```

Indien de diskette nog niet is geformatteerd, resulteert dat in voorgaande mededeling. Staan er één of meer programma's op de diskette, dan zullen die worden weergegeven op het beeldscherm. Het commando 'FILES' wordt bij de commando's in paragraaf 4.4 uitgebreid besproken.

4.3 Formatteren

Het commando CALL FORMAT zorgt ervoor dat de diskette gereed wordt gemaakt om programma's, bestanden en files te ontvangen (zie ook paragraaf 1.1). Alle nieuwe diskettes dient men dan ook voor het gebruik te *formatteren*.
Diskettes kunnen zowel onder MSX Disk BASIC als onder MSX-DOS worden geformatteerd. De onder MSX-DOS geformatteerde diskettes zijn tevens geschikt voor Disk BASIC. Evenzo zijn diskettes geformatteerd onder Disk BASIC geschikt voor MSX-DOS. We gaan nu eerst een aantal diskettes gereed maken voor gebruik:

```
Ok
CALL FORMAT
```

Na het geven van een <RETURN> verschijnt de volgende vraag op uw beeldscherm:

```
Drive name? (A,B) ■
```

Heeft men één diskdrive dan is diskdrive A: het meest gebruikelijk om te kiezen voor het formatteren. Na het intoetsen van de letter A, komt de computer voor alle zekerheid nog met de volgende mededeling:

```
Strike a key when ready ■
```

Indien de te formatteren diskette nog niet is geplaatst, kunt u dat alsnog doen.
Bij het intoetsen van de letter 'B' zal de computer vragen om diskette B te plaatsen:

```
Insert diskette for drive B:
and strike a key when ready■
```

Men benoemt nu niet de diskdrive maar de diskette. Heeft u méér dan twee diskdrives (dus twee interfaces), dan zal de eerste vraag luiden:

```
Drive name? (A,B,C,D) ■
```

Na het plaatsen van de te formatteren diskette en het indrukken van de juiste letter voor de diskdrive (bij meer dan één diskdrive) waarin die diskette is geplaatst, kunt u een willekeurige toets indrukken en zal de diskette worden geformatteerd. Disk BASIC meldt u dat het formatteren is voltooid met de volgende boodschap:

```
Format complete
Ok
```

Het is verstandig om vooraf een aantal diskettes te formatteren.
Bij het gebruik van één diskdrive is het beter de nieuwe diskette eerst in de diskdrive te plaatsen voor u het commando CALL FORMAT geeft, dit om eventuele vergissingen te voorkomen.
Heeft u bij het gebruik van één diskdrive de letter B: ingetoetst na het FORMAT-commando, zal de computer om diskette A vragen na het eerst volgende commando.
Bij meer diskdrives zal na het formatteren de default diskdrive weer zijn geactiveerd.
In plaats van het commando 'CALL FORMAT', mogen we ook de verkorte versie '–FORMAT' gebruiken, tevens accepteert de computer zowel hoofdletters als kleine letters in de commando's.

4.4 MSX Disk BASIC-commando's

Er zijn veel MSX BASIC-commando's die tevens betrekking hebben op het gebruik van de diskdrive, zoals LOAD, SAVE. De betekenis en het gebruik van dat soort commando's kunt u terugvinden in het handboek van uw computer. Hier behandelen we alleen de 'extra' en specifieke commando's voor gebruik bij een diskdrive. U dient er wel aan te denken dat het benoemen van een andere diskette binnen de aanhalingstekens moet gebeuren direct voor de filenaam.

APPEND *Wordt gebruikt om nieuwe data aan een reeds bestaand sequentieel bestand toe te voegen.*

Voorbeeld: `10 OPEN "DATA" FOR APPEND AS #1`

Het reeds bestaande bestand 'DATA' wordt geopend voor het toevoegen van gegevens. Het wegschrijven van gegevens naar het DATA-bestand, is verder gelijk aan de methode voor het creëren van een sequentieel bestand. Zie voor meer gegevens omtrent het maken van sequentiële bestanden hoofdstuk 5.

COPY TO *Dit commando dient voor het kopiëren van één of meer files van de ene naar de andere diskette. Dit commando werkt zowel voor één diskdrive als twee diskdrives.*

Voorbeeld: `COPY "A:TESTPROG.BAS" TO "B:TEST.BAS"`

Dit commando zal het programma TESTPROG.BAS kopiëren van diskette A: naar diskette B:, maar onder een andere naam, namelijk TEST.BAS. Een programma of file kopiëren op dezelfde diskette onder dezelfde naam is mogelijk, doch moet in de praktijk worden vermeden, daar bij lange files de informatie gedeeltelijk verloren kan gaan. Na het kopiëren van programma's, bestanden of files krijgt u onder het COPY-commando de mededeling 'Ok'.
De tweede naam achter 'TO' is een optie en kan worden vervangen door alléén een letter om daarmee een andere diskette of diskdrive aan te geven, waardoor het programma onder dezelfde naam naar de opgegeven diskette wordt gekopieerd.

```
COPY "TESTPROG.BAS" TO "B:"
```

zal dus het programma TESTPROG.BAS kopiëren van diskette A: naar diskette B:.
Ervan uitgaande dat u voorlopig maar één diskdrive heeft,

krijgt u na het inlezen van het programma 'TESTPROG.BAS' de volgende mededeling op het beeldscherm:

```
Insert diskette for drive B:
and strike a key when ready■
```

Direct na het indrukken van een willekeurige toets, zal het programma worden weggeschreven (natuurlijk wel eerst diskette B: plaatsen!). Na het wegschrijven vraagt de computer om diskette A: met de volgende mededeling:

```
Insert diskette for drive A:
and strike a key when ready■
```

Na het wisselen van de diskettes voor het kopiëren zal de computer u weer met 'Ok' melden dat hij klaar is met kopiëren.
Andere combinaties zijn natuurlijk ook mogelijk, om bijvoorbeeld vanaf een andere diskette naar diskette A: te kopiëren, kunt u het volgende commando gebruiken:

```
COPY "C:TESTPROG.BAS"
```

Het programma TESTPROG.BAS wordt nu vanaf diskette C gekopieerd naar diskette A:. Indien men van een bepaalde diskette naar een andere bepaalde diskette wilt kopiëren, dient men beide diskettes of diskdrives op te geven.

```
COPY "C:TESTPROG.BAS" TO "D:"
of
COPY "C:TESTPROG.BAS" TO "D:TEST.BAS"
```

In het eerste geval wordt het programma gekopieerd en onder dezelfde naam weggeschreven naar diskette D. In het tweede geval zal het programma ook naar diskette D worden gekopieerd, maar krijgt het nu de naam TEST.BAS.

BASIC-programma's kunnen tekstfiles maken die dan moeten worden geSAVEd met de A-optie. Het einde van zo'n tekst- of ASCII-file wordt altijd gekenmerkt door een '<CONTROL>-Z'-(^Z) code; dat is de MSX-karaktercode 26.
Binaire files en programma's hebben geen ^Z-code aan het einde.
Het SAVEn met de A-optie gaat als volgt:

```
SAVE "TEST.ASC",A
```

Nu wordt het weggeschreven programma of de tekstfile op de diskette afgesloten met een ^Z. Alle gebruikte tekens in het pro-

gramma worden als normale ASCII-karakters weggeschreven. Gebruikt men deze A-optie niet bij het SAVEn, dan wordt het programma in gecomprimeerde vorm weggeschreven. Dit comprimeren gebeurt met alle BASIC-woorden, stukken tekst binnen een programma blijven natuurlijk tekst! Het voordeel van een ASCII-file is het kopiëren naar andere aangesloten apparaten en het samenvoegen van ASCII-files met het commando 'MERGE' bij BASIC en onder MSX-DOS met het COPY-commando.

Bij het kopiëren is het gebruik van wildkarakters toegestaan, bijvoorbeeld:

```
COPY "*.BAS" TO "*.TEX"
```

Alle files voorzien van de extension .BAS worden nu gekopieerd naar nieuwe files met dezelfde naam doch een andere extension (.TEX) en wel op dezelfde diskette. De files die worden gekopieerd zullen ook als afzonderlijke files op de diskette blijven staan.
Op deze wijze kunnen we ook in één keer een gehele diskette kopiëren:

```
COPY "*.*" TO "B:"
```

Alle files op diskette A worden nu overgebracht naar diskette B. Ook het gebruik van het vraagteken als wildkarakter is mogelijk:

```
COPY "TEKST?.TXT" TO "TEKST?.TTT"
```

Met dit commando zullen alle files die beginnen met 'TEKST' en daarachter nog één karakter hebben en eindigend op .TXT worden gekopieerd naar nieuwe files met dezelfde naam, maar met een andere extension. De volgende files zullen bijvoorbeeld worden gekopieerd:

TEKST1.TXT	wordt	TEKST1.TTT
TEKST3.TXT	wordt	TEKST3.TTT
TEKSTA.TXT	wordt	TEKSTA.TTT
maar ook ...		
TEKST.TXT	wordt	TEKST.TTT

In de inhoudsopgave van de diskette worden alle filenamen opgeslagen in de vorm van acht karakters vóór de extension. De file 'TEKST.TXT' staat in de inhoudsopgave als 'TEKST   .TXT', tussen 'TEKST' en '.' bevinden zich nog drie spaties! Het gegeven commando ziet op de plaats van het vraag-

teken achter TEKST een spatie (MSX-karaktercode 32) waardoor de file voldoet aan dat commando en ook wordt gekopieerd. Gebruik daarom bij het kopiëren nooit één of meer vraagtekens aan het einde van een filenaam.

COPY TO *Dit commando kan tevens worden gebruikt om ASCII-files te kopiëren naar andere apparatuur.*

Voorbeeld: `COPY "TEKST.TXT" TO "CON"`

Dit commando zal de inhoud van de file TEKST.TXT kopiëren naar CON (het CONsole, in dit geval het beeldscherm). U krijgt op het beeldscherm de tekst te zien waaruit deze file bestaat. In plaats van CON kunt u ook andere apparatuur invullen (AUX) en natuurlijk de printer, bijvoorbeeld:

```
COPY "TEKST.TXT" TO "LST"
```

zal de tekstfile naar de printer sturen en alle informatie op papier zetten. In plaats van de aanduiding 'LST' kan men ook 'PRN' gebruiken.

```
COPY "CON" TO "A:TEST.TES"
```

Dit commando zal alles dat u via het CONsole (in dit geval het toetsenbord) invoert, overbrengen naar diskette A: onder de naam TEST.TES. Dit gebeurt na het afsluiten van de tekst met een ^Z-code daar het om een tekst- of ASCII-file gaat. Meer omtrent het overbrengen van teksten en commandoprogramma's naar de diskette vindt u in het hoofdstuk over 'BATCH'-programma's.
Natuurlijk is het ook mogelijk om direct vanaf het toetsenbord alles over te brengen naar de printer.

```
COPY "CON" TO "LST"
```

Dit wordt direct na het afsluiten door de printer op papier gezet.

```
^Z
Ok
```

Vergeet hier niet de <CONTROL>-Z als teken dat de ASCII-file wordt afgesloten.

FIELD *Dit commando dient voor het creëren van ruimte in een buffer voor variabelen ten behoeve van een Random Access File.*

Voorbeeld:
```
FIELD #1,20 AS NAAM$,30 AS
ADRES$,7 AS POSTCODE$,20 AS
PLAATS$,11 AS TELEFOON$
```

Hiermee worden de velden binnen een 'record' van een Random Access File gedimensioneerd. Het eerste veld krijgt ruimte voor 20 karakters en de variabele 'NAAM$', het tweede veld heeft 30 karakters en de variabele heet ADRES$, enz. In totaal heeft het record 88 posities; dit moet voor het gebruik eerst worden gedimensioneerd (zie het LEN-commando).
Bij Random Acces Files kan men het eerste record het beste gebruiken om het aantal records te registreren waaruit de file bestaat (zie verder hoofdstuk 5).

FILES en LFILES *Deze commando's tonen u alle 'files' (inhoudsopgave) van de diskette op het beeldscherm, respectievelijk via de printer.*

Voorbeeld: `FILES`

U krijgt nu alle filenamen op uw beeldscherm te zien voorzien van hun extension.

```
FILES "B:"
```

toont alle files van diskette B:. Bij het gebruik van één diskdrive zal de computer u vragen om de andere diskette te plaatsen. Het is ook mogelijk om files met een bepaalde extension weer te geven:

```
FILES "*.<extension>"
```

Dit commando toont alle files met dezelfde (opgegeven) extension. Het gebruik van het vraagteken als wildkarakter is ook toegestaan. Hier gelden dezelfde regels als bij het COPY-commando.
Om een uitdraai van de inhoudsopgave op papier te krijgen, geeft men het commando:

```
LFILES
```

Met dit commando gaan de filenamen naar de printer en niet naar het beeldscherm. Alle mogelijkheden van het commando 'FILES' zijn ook van toepassing op dit commando, dus het gebruik van wildkarakters of de benoeming van een andere diskette zijn toegestaan.

FORMAT *Dit commando maakt de diskette gereed voor het ontvangen en opslaan van programma's e.d. onder MSX Disk BASIC.*

Voorbeeld: `CALL FORMAT`
of `_FORMAT`

Na dit commando zal de computer de volgende vraag stellen:

```
Drive name? (A,B) ■
```

Zijn er twee diskdrive-interfaces op uw computer aangesloten, dan zal de vraag luiden:

```
Drive name? (A,B,C,D,) ■
```

Na het opgeven van de gewenste diskette of diskdrive zal de volgende boodschap verschijnen:

```
Strike a key when ready ■
```

Nadat u de juiste diskette in de juiste diskdrive heeft geplaatst, kunt u een willekeurige toets indrukken en zal de diskette worden geformatteerd. Zodra het formatteren gereed is, meldt de computer dit:

```
Format complete
Ok
```

Diskettes die u onder MSX-DOS heeft geformatteerd, kunt u zonder meer gebruiken voor Disk BASIC.

GET *Dit commando dient voor het lezen van een record uit een Random Access File dat daarna wordt opgeslagen in een random filebuffer.*

Voorbeeld: `GET#1,12`

Nadat de file is geopend kan men met dit commando een record inlezen. Direct achter het GET-commando komt eerst het 'logische filenummer' waar de Random Access File mee werd geopend. Achter de komma staat het recordnummer. Het in dit voorbeeld genoemde commando leest dus van file '#1' het twaalfde record.
Geeft men geen recordnummer op, dan zal het GET-commando het eerstvolgende record lezen.
Na een GET-commando kan men met INPUT# en LINE INPUT# de gegevens uit de random filebuffer lezen. Zie voor meer bijzonderheden hoofdstuk 5.

KILL *Commando voor het wissen van één of meer files van de diskette. Achter het commando wordt de te wissen file gespecificeerd.*

Voorbeeld: `KILL "TESTPROG.BAS"`

De file TESTPROG.BAS wordt van de diskette gewist. Zolang er geen drive wordt opgegeven, zal dat altijd van diskette A: zijn. Door voor de filenaam de diskette of diskdrive in te toetsen, wordt de file van de genoemde diskette gewist, bijvoorbeeld:

```
KILL "B:TESTPROG.BAS"
```

Het gebruik van wildkarakters is hier toegestaan, hoewel men daar erg voorzichtig mee moet zijn.

```
KILL "B:*.TXT"
```

Deze opdracht zal alle files met de extension .TXT die zich op diskette B bevinden, wissen. Om alle files ongeacht hun naam of extension van een diskette te wissen, kunnen we dus het volgende commando gebruiken:

```
KILL "B:*.*"
```

Denk eraan dat de computer deze opdracht zonder blikken of blozen uitvoert! U bent dan alle programma's, bestanden en files kwijt die op diskette B: stonden.
Gebruikt u dit commando terwijl er een sequentieel bestand of Random Acces File is geopend, dan zal de volgende mededeling op uw beeldscherm verschijnen:

```
File still open
Ok
```

Er kan in dit geval niets van de diskette worden gewist.

LEN *Dit commando dient voor het dimensioneren van de recordlengte bij Random Access Files.*

Voorbeeld: `10 OPEN "A:ADRES.BES" AS #1 LEN=75`

Met deze programmaregel wordt op diskette A: het BEStand ADRES geopend onder het logische bestandsnummer #1 en krijgt ieder record een lengte van 75 karakters.
Hierna dient men de afzonderlijke velden binnen de records te dimensioneren met het commando FIELD.
Zie voor meer gegevens omtrent het werken met Random Access Files hoofdstuk 5.

LSET en RSET *Commando's om de gegevens vanuit het werkgeheugen naar de filebuffer te schrijven.*

Voorbeeld: `50 LSET NAAM$=N$`

De variabele N$ wordt opgeslagen in de gereserveerde ruimte NAAM$ die zich in de filebuffer bevindt en wel aan de linkerzijde van de gereserveerde ruimte. Met andere woorden: de stringvariabele N$ wordt opgeslagen in NAAM$ als een reeks karakters die bestaat uit N$ aangevuld met spaties.

Voorbeeld: `50 RSET NAAM$=N$`

De variabele N$ wordt nu aan de rechterzijde opgeslagen in de gereserveerde ruimte en staan er spaties vóór N$. Beide commando's behoren tot bewerkingen die nodig zijn bij de Random Access Files, zie verder hoofdstuk 5.

NAME AS *Verandert de filenaam (Engels reNAME) van een te specificeren file in een nieuw gespecificeerde filenaam.*

Voorbeeld: `NAME "TEKST.TXT" AS "TEKST1.TXT"`

De bestaande file TEKST.TXT krijgt de nieuwe naam TEKST1.TXT. De inhoud van de file TEKST.TXT blijft ongewijzigd en op dezelfde plaats op de diskette staan; alleen in de inhoudsopgave wordt de naam veranderd. Staat de file die van naam moet veranderen niet op de default diskette, dan dient men voor de eerste filenaam de juiste diskette of diskdrive op te geven:

```
NAME "B:TEKST.TXT" AS "TEKST1.TXT"
```

De file op diskette B krijgt nu een andere naam. Het opgeven van een diskdrivenaam voor de tweede filenaam zal worden genegeerd; de file wordt *altijd* op de default diskette van naam veranderd, tenzij men voor de eerste filenaam een andere diskette of diskdrive specificeert.
Het is toegestaan om wildkarakters te gebruiken bij het veranderen van filenamen.

```
NAME "*.TXT" AS "*.TST"
NAME "ABCDE" AS "B??B?"
```

In het eerste geval zullen alle filenamen met de extension .TXT worden vervangen door dezelfde filenamen, maar nu met de extension .TST.

Bij het tweede commando zal de file ABCDE veranderen in BBCBE. Op de plaats van het wildkarakter (?) worden dezelfde letters gehandhaafd. Op de plaatsen waar een letter staat, wordt de oude letter vervangen door de nieuw opgegeven letter. Let er op dat in de inhoudsopgave de filenamen *altijd* uit acht karakters bestaan; de naam wordt met spaties aangevuld tot acht karakters. Gebruik daarom nooit vraagtekens als wildkarakters aan het einde van een filenaam.
Geeft men voor de nieuwe file een filenaam op die overeenkomt met een reeds op de diskette aanwezige file, dan zal er een foutmelding op het beeldscherm verschijnen:

```
File already exists
```

Als we een niet op de diskette staande file van naam willen veranderen, krijgen we op het beeldscherm de volgende melding:

```
File not found
```

PUT *Hiermee schrijven we de inhoud van de filebuffer naar de diskette.*

Voorbeeld: `80 PUT #1, R%`

Nu worden alle gegevens die zich in de filebuffer bevinden naar de diskette geschreven en wel naar het bestand met nummer '#1' en naar 'record' nummer 'R%'. R% is in dit geval een variabele maar kan ook een normaal getal zijn!
In hoofdstuk 5 wordt dit commando verder behandeld bij het creëren en werken met Random Access Files.

SYSTEM *Specifiek MSX-DOS-commando dat alleen werkt bij MSX Disk BASIC.*

Voorbeeld: `CALL SYSTEM`
of `_SYSTEM`

Als men vanuit MSX-DOS met het commando 'BASIC' is overgegaan naar Disk BASIC, kan men zonder de computer uit te zetten of te 'resetten' met één van deze twee commando's terugkeren naar het MSX-DOS-besturingssysteem. De systeemdiskette voorzien van de files MSX-DOS.SYS en COMMAND.COM dient eerst te worden geplaatst in diskdrive A. De commando's werken niet als er is opgestart in BASIC of Disk BASIC.

VERIFY *Met dit commando kan alle data die naar de diskette wordt weggeschreven, worden gecontroleerd.*

Voorbeeld: `CALL VERIFY ON`
of `_VERIFY ON`

Nu wordt de verify-mode (controleren) ingeschakeld. Alle data die naar de diskette wordt gestuurd zal, nadat het op de diskette staat, weer worden gelezen en gecontroleerd aan de hand van de inhoud van de computer. Indien er een fout is opgetreden bij het wegschrijven naar de diskette, verschijnt er een foutmelding op het beeldscherm:

```
DISK I/O error
```

Het uitzetten van de verify-mode gaat met het commando:

`CALL VERIFY OFF`
of `_VERIFY OFF`

Het wegschrijven van gegevens kost nu meer tijd maar men heeft dan wel controle over de juistheid van de weggeschreven gegevens.
Na het opstarten van **MSX** Disk BASIC staat de verify-mode altijd op OFF.

4.5 MSX Disk BASIC-functies

Buiten de specifieke Disk BASIC-commando's kent Disk BASIC nog een aantal functies die na het oproepen een bewerking uitvoeren. Deze functies dienen hoofdzakelijk voor het lezen en schrijven in Random Access Files. In het volgende hoofdstuk zullen deze functies uitgebreid aan de orde komen.

CVI – CVS – CVD *Deze functies zetten de inhoud van een 'numerieke string' om in een numerieke waarde. Deze numerieke string dient te zijn gemaakt met de functies MKI$, MKS$ of MKD$.*

Voorbeeld:

```
60 X=CVI (TEL$)
70 Y=CVS (A$)
80 Z=CVD (B$)
```

CVI verwerkt een 2-bytes string en zet die om naar een integer (geheel getal). Omdat het een 2-bytes string is, kan de integer niet groter zijn dan '32767' of kleiner dan '–32768'. Om dit duidelijk te maken laten we zien hoe een 2-bytes integer wordt opgeslagen in de computer:

BCD	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
decimaal	↑ 7 × 4096	15 × 256	15 × 16	15 × 1
	positief of negatief			

Het eerste bit geeft aan of het een positief dan wel een negatief getal is. De overige bits geven de waarde van het getal aan. Daar de eerste nibble het voorste bit niet mag gebruiken, kunnen we niet hoger komen dan '7'. De maximale waarde wordt dus:

$$7 \times 4096 + 15 \times 256 + 15 \times 16 + 15 \times 1 = 32768!$$

CVS verwerkt een 4-bytes string tot een waarde met 'enkele' precisie. Uit deze bewerking kunnen getallen met maximaal zes cijfers voortkomen, hetgeen we weer duidelijk maken aan de hand van een voorbeeld.

BCD	1e byte	2e byte	3e byte	4e byte
		2 cijfers	2 cijfers	2 cijfers

- aantal cijfers voor de komma
- positief of negatief

Het eerste byte geeft aan of het getal positief of negatief is en hoeveel cijfers er voor de komma staan. De overige bytes geven steeds twee cijfers, waardoor het getal dus maximaal uit zes cijfers kan bestaan.

CVD verwerkt een 8-bytes string tot een waarde met 'dubbele' precisie. De functie werkt hetzelfde als CVS, doch gebruikt acht bytes voor de opslag van het getal in plaats van vier bytes. Het eerste byte wordt weer gebruikt voor het aangeven van een positieve of negatieve waarde en voor het aangeven van de exponent. De overige zeven bytes geven ook hier steeds twee cijfers, waardoor het getal maximaal uit 14 cijfers kan bestaan.

DSKF *Deze functie leest uit de inhoudsopgave de vrije opslagruimte die zich nog op de diskette bevindt.*

Voorbeeld: `PRINT DSKF (diskdrive nummer)`

Het diskdrive nummer dient tussen haakjes te staan en correspondeert met:

0	default diskdrive
1	diskdrive A:
2	diskdrive B:
enz.	

Het getal dat deze functie van de diskette leest en op het beeldscherm weergeeft betreft kilobytes. Een waarde van 179 wil dus zeggen een beschikbare ruimte van 179 × 1024 = 183296 bytes voor opslag van gegevens.

LOC *Deze functie geeft het laatst gelezen of geschreven recordnummer aan bij een Random Access File. Bij een sequentieel bestand zal de functie het totale aantal ingelezen of weggeschreven records aangeven.*

Voorbeeld: `PRINT LOC (bestandsnummer)`

Het nummer dient overeen te komen met het bestandsnummer van het geopend bestand. Is er nog niet gelezen of geschreven in een geopende Random Access File, dan krijgt men de waarde 0. Bij een sequentieel bestand krijgt men in dit geval de waarde 1, omdat bij het openen van een sequentieel bestand het eerste record altijd wordt gelezen. Probeert men deze functie zonder dat het genoemde bestand is geopend, dan resulteert dat in de melding:

```
File not OPEN
```

LOF *Deze functie geeft de totale lengte van een bestand in bytes.*

Voorbeeld: `PRINT LOF (bestandsnummer)`

Denk eraan dat ieder gegeven in een sequentieel bestand wordt afgesloten met twee codes, tevens heeft het bestand nog een begincode. Het getal dat deze functie genereert bestaat uit de hoeveelheid ingevoerde karakters plus één beginkarakter en twee karakters extra per ingevoerd 'item'.

MKI$ – MKS$ – MKD$ *Deze functies zetten een numerieke waarde om in een numerieke string voor opslag in een Random Access File.*

Voorbeeld:

```
60 TEL$=MKI$(A%)
70 A$=MKS$(B!)
80 B$=MKD$(C#)
```

MKI$ zet integers tussen de '32767' en '–32768' om in een numerieke string die alleen gelezen kan worden met CVI.

MKS$ zet een getal met enkele precisie om naar een numerieke string, hierbij wordt het getal indien nodig afgerond op zes cijfers. Het afronden gebeurt op de normale manier: 123,4567 wordt 123,457 en 123,4562 wordt 123,456. De numerieke waarde kan worden gelezen met de CVS-functie.

MKD$ zet getallen met dubbele precisie om die met de CVD-functie kunnen worden terug gelezen.

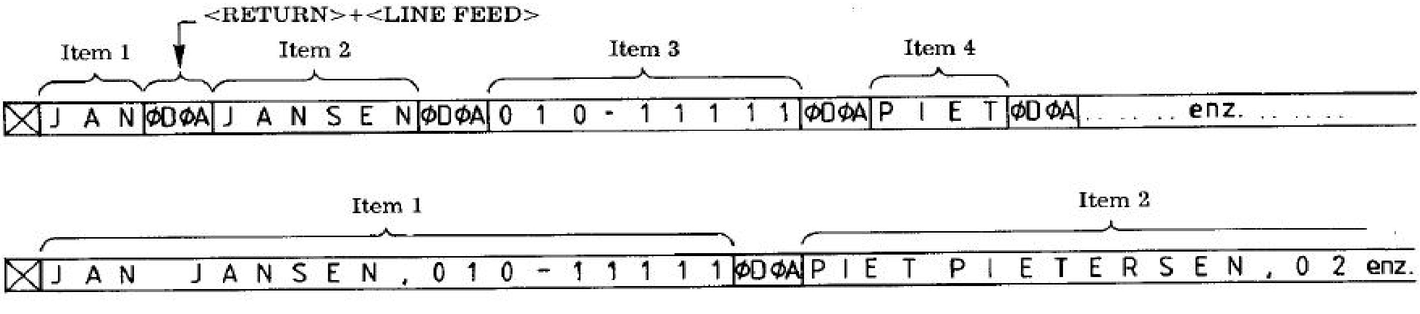
Het voordeel van deze functies is de ruimtebesparing, MKI$ beslaat slechts twee bytes, MKS$ vier bytes en MKD$ totaal acht bytes.

HOOFDSTUK 5

WERKEN MET BESTANDEN

5.1 Opbouw van een bestand

Om met bestanden te werken is het van belang om het begrip bestand en het manipuleren met de gegevens goed onder de knie te krijgen. Als men namelijk de opbouw van een bestand niet goed begrijpt, is het werken met zo'n bestand haast ondoenlijk. We beginnen met de opbouw van een sequentieel bestand. Een sequentieel bestand noemt men ook wel eens een 'volgorde bestand' omdat alle gegevens in een vaste volgorde worden opgeslagen. Deze volgorde wordt bepaald door de volgorde van invoeren. De ingevoerde gegevens worden achter elkaar weggeschreven naar het bestand op de diskette. De verschillende 'items' worden van elkaar gescheiden door een <RETURN> en een <LINE FEED>, respectievelijk de MSX-karaktercodes 13 en 10; hexadecimaal 0D en 0A. Bekijken we nu de werkelijke opbouw van het sequentiële bestand, dan zien we het volgende:



Afbeelding 5.1 De opbouw van een sequentieel bestand, men kan de afzonderlijke items wegschrijven, maar ook de bij elkaar behorende gegevens als één item gescheiden door een komma.

We zien al direct dat de eerste manier van wegschrijven meer bytes in beslag neemt dan de tweede. Tevens zien we dat de lengte van één item afhankelijk is van het aantal karakters van dat item en de manier van wegschrijven naar het bestand.
De gegevens in een sequentieel bestand kennen dan ook geen vaste lengte en de verschillende items kunnen alleen worden onderscheiden door de twee codes (0D en 0A) die het einde van een item aangeven.

Aan de opbouw van het bestand zien we al direct dat een sequentieel bestand de enige vorm van gegevensopslag is die we met een cassetterecorder kunnen verwezenlijken. Het bestand wordt dan vanaf de cassette gelezen tot de filebuffer vol is en de computer laat de cassetterecorder stoppen. Zijn alle items uit de buffer gelezen, dan wordt het vervolg vanaf de cassette in de computer gebracht, net zo lang tot er geen gegevens meer zijn.

Voor we gegevens naar een bestand kunnen schrijven, dienen we het bestand eerst te openen. Om de verschillende commando's beter te leren kennen, bespreken we hier alleen die gedeelten die van belang zijn voor de opbouw van een bestand. In Appendix F staat een compleet programma voor een 'telefoonboek'-bestand waarin u de verschillende commando's kunt terugvinden.

5.2 Openen van een bestand en invoeren van de gegevens

Het openen van het bestand gaat op dezelfde wijze als bij het gewone MSX BASIC:

```
10 OPEN "A:TESTEN.DAT" FOR OUTPUT AS #1
```

(A: = apparaat; TESTEN.DAT = bestandsnaam; OUTPUT = mode; #1 = bestandsnummer)

In de computerwordt de filebuffer opgezet voor het bestand 'nummer #1'. De computer heeft daar een speciale ruimte voor gereserveerd, wil men met meer bestanden tegelijkertijd werken, dan dient men ook meer filebuffers te reserveren. Zie hiervoor het MAXFILES-commando in uw handboek.
Bij het openen wordt tevens de naam van het bestand in de inhoudsopgave van de diskette geschreven en een eerste sector gereserveerd.
De gekozen 'mode' is 'OUTPUT' (uitvoeren), hetgeen wil zeggen wegschrijven. Daarachter vinden we het bestandsnummer, over het algemeen is dat #1. Dit nummer is nodig om in het geval van meer bestanden deze van elkaar te kunnen onderscheiden, bijvoorbeeld bestand #1, bestand #2 en bestand #3. Deze drie kunnen dan tegelijkertijd 'open' zijn om uit te lezen of in te schrijven.
Geheel vooraan in de opdracht vinden we de benaming van het apparaat om aan te geven waar de gegevens naar toe moeten; daarachter de bestandsnaam waaronder het bestand wordt weggeschreven naar de diskette. Alleen de computer houdt bij dat de gekozen bestandsnaam na het openen behoort bij filebuffer #1.
Nu gaan we de gegevens vanaf het toetsenbord invoeren en voorlopig opslaan in de filebuffer.

```
20 INPUT"Naam:";NA$
30 IF NA$="*" THEN 90
40 INPUT"Telefoon:";TE$
50 INPUT"Is dit correct";KN$
60 IF KN$="j" OR KN$="J" THEN 70 ELSE 80
70 PRINT#1,NA$;",";TE$
80 GOTO 20
90 CLOSE #1
```

In het uiteindelijke programma (Appendix F) staan een aantal opdrachten om beeldschermregels te wissen, die verder niet van belang zijn voor de werking van het programma; ze dienen alleen ter verfraaiing. In regel 20 vraagt het programma om de invoer van een naam. Denk eraan dat men alle namen of in kleine of in hoofdletters moet opgeven. Gebruik deze niet door elkaar, doet u dit wel probeer dan altijd hetzelfde stramien aan te houden bij het invoeren van gegevens. Vervolgens wordt gecontroleerd of het 'einde-invoerteken' (*) werd getypt om in dit geval naar regel 90 te springen en het bestand te sluiten. Bestaat variabele NA$ uit een ander karakter of karakters, dan gaat het programma naar de volgende regel en vraagt om het bijbehorende telefoonnummer. Ten slotte heeft u de gelegenheid om de gegevens op uw beeldscherm te controleren en indien nodig te wijzigen.
Na het akkoord bevinden van de gegevens zorgt regel 70 ervoor dat de inhoud van de variabelen NA$ en TE$ naar de gereserveerde ruimte van filebuffer '#1' gaat in de volgorde zoals aangegeven in afb. 5.1 (onderste tekening). Daarna wordt er teruggesprongen naar regel 20 en kan men de volgende naam en het telefoonnummer invoeren. Deze routine blijft in een lus rondgaan tot men geen naam, maar de asterisk intoetst en de routine via regel 30 naar het eind van het programma springt.In regel 90 bevindt zich het CLOSE-commando dat ervoor zorgt dat de gegevens vanuit filebuffer #1 naar diskette worden geschreven. De filebuffer heeft een capaciteit die afhankelijk is van het aantal filebuffers, hetgeen vaak betekent dat de inhoud van de buffer bij een grote hoeveelheid gegevens tussentijds moet worden weggeschreven.

5.3 Lezen van het bestand

We willen natuurlijk de inhoud van het zojuist gecreëerde bestand controleren. Hiertoe maken we een klein programma om het bestand te kunnen lezen:

```
10 OPEN "A:TESTEN.DAT" FOR INPUT AS #1
                           mode
```

Als mode kiezen we 'INPUT' (invoer) waardoor de gegevens vanaf de diskette 'in' de computer worden gebracht, in filebuffer '#1'.
Men had hier ieder gewenst bestandsnummer kunnen kiezen, het is niet zo dat een bestand dat onder nummer '#1' is weggeschreven ook onder dat nummer moet worden gelezen! Wel dient men het gekozen bestandsnummer na het openen van dat bestand in het gehele programma gelijk te houden. In het hiervoor genoemde statement behoort bestandsnummer #1 bij de gegevens in het bestand met de naam 'TESTEN.DAT'. Nadat het bestand is gesloten kunnen we bestandsnummer #1 voor andere bestanden gebruiken.
Direct na het uitvoeren van regel 10 worden de eerste '256' bytes overgebracht naar de buffer, waarna het lezen in de buffer kan beginnen.

```
20 INPUT #1,NA$,TE$
30 T=T+1
40 PRINT NA$,TE$
50 IF T<10 THEN 20
60 . . .
```

Dit korte programma zal de eerste tien 'items' lezen en op het beeldscherm weergeven. Regel 30 telt het aantal weergegeven regels en regel 50 controleert of dat kleiner dan tien is.
Regel 20 leest de twee variabelen van het eerste item in. Let er op dat hier het commando 'INPUT #1' dient te worden gebruikt om de gegevens uit de buffer te lezen (zie uw handboek voor bijzonderheden).

5.4 Toevoegen aan het bestand

Als het mogelijk is, willen we achteraf natuurlijk wel iets kunnen toevoegen aan het bestand. Hiervoor heeft het OPEN-commando de 'mode' APPEND:

```
10 OPEN "A:TESTEN.DAT" FOR APPEND AS #1
                           ------
                            mode
```

Het desbetreffende bestand wordt nu vanaf de diskette gelezen en in de filebuffer opgeslagen. Bestaat het bestand inmiddels uit meer sectoren op de diskette, dan zal de laatste sector worden gelezen! Hierachter kunnen de aanvullingen worden uitgevoerd op dezelfde wijze als bij het openen van een nieuw bestand. In het uiteindelijke programma ziet u dan ook een GOSUB naar de invoer-routine. Hier geldt ook weer dat de asterisk 'einde'-invoer betekent.

Meer over de werking en het gebruik van sequentiële bestanden vindt u in het handboek van de computer. De werking en commando's voor cassette en diskette zijn gelijk.
Het sequentieel bestand werkt over het algemeen goed en geeft weinig problemen. Een nadeel is dat men het gehele bestand moet 'doorlezen' voordat men gevonden heeft wat men zoekt, als dat gezochte toevallig helemaal aan het einde staat. Daarnaast is ook het sorteren van een dergelijk bestand nogal omslachtig. En corrigeren is helemaal een probleem op zich. Vandaar dat men meestal de Random Access Files gebruikt bij het manipuleren met bestanden.
Toch wil ik hier wel een paar tips geven over het werken met sequentiële bestanden.
Om een bestand te corrigeren, kan men bijvoorbeeld twee bestanden openen, één (het orginele) om in te lezen met het commando INPUT #1 en een tweede (het gecorrigeerde) om weg te schrijven met het commando PRINT #2. Nu kunnen we 'item' voor 'item' uit de buffer halen en op het scherm laten weergeven. Willen we de gegevens corrigeren, dan doen we dat, zo niet dan gaan de gegevens zo door naar de tweede buffer met het commando PRINT #2. Op deze manier lezen we alle items één voor één en corrigeren zonodig of sturen ze rechtstreeks door naar de tweede buffer. Ten slotte sluiten we beide bestanden.
Om een sequentieel bestand te sorteren, op alfabet of numeriek, kunnen we net zo te werk gaan. We lezen een aantal 'items' in, bijvoorbeeld zeven stuks en gaan die eerst sorteren. We schrijven nu de eerste zeven gesorteerde items naar buffer #2. We lezen de volgende zeven items in, sorteren deze en schrijven ze nu naar buffer #2. Zo gaan we door tot aan het einde van het bestand.
Het sorteren is nu natuurlijk nog niet achter de rug, we dienen de gehele procedure nogmaals te herhalen, maar nu lezen we negen items in en sorteren deze, nu weer wegschrijven en de volgende negen inlezen enz. Zo zal men een aantal keren de bestanden door de computer moeten halen met als eindresultaat een gesorteerd sequentieel bestand.

5.5 Random Access Files

Een Random Access File noemt men ook wel een direct toegankelijk bestand. In een dergelijk bestand werken we met 'records' of kaarten. Deze kaarten zijn ingedeeld volgens een vast patroon, deze verdeling noemen we in computertaal 'velden'. Een record bevat dus één of meer velden die onderling van grootte kunnen verschillen. Alle records zijn net als bij kaarten in een echte kaartenbak even groot.
Kijken we naar een normaal adressenbestand, dan kunnen we ons de volgende indeling voorstellen:

Naam
Adres
Postcode
Woonplaats
Telefoonnummer
Bijzonderheden

Voor we een definitieve indeling gaan maken dienen we eerst het gebruik van zo'n direct toegankelijk bestand onder de loep te nemen. Slaan we alleen de gegevens in dit bestand op en gebruiken we het verder slechts als naslagwerk, dan heeft het weinig zin om zo'n bestand te maken.
Zijn het adressen van leden van een vereniging, dan kunnen we ons voorstellen dat er af en toe adressen moeten worden geprint om een verenigingsblad te verzenden. Ook het uitprinten van een ledenlijst behoort tot de regelmatig voorkomende werkzaamheden.
Bestaat de vereniging uit zowel vrouwelijke als mannelijke leden, dan is voor het verzenden van het verenigingsblad de juiste aanhef geen overbodige luxe. Willen we op naam kunnen sorteren, dan is een 'veld' met alleen de achternaam ook handig, vandaar de volgende indeling:

VELDNAAM	LENGTE	VARIABELE
Voorvoegsel	3	BE$
Voorletters	8	VO$
Tussenvoegsel	6	TU$
Achternaam	18	NA$
Adres	25	AD$
Postcode	7	PO$
Woonplaats	20	PL$
Telefoonnummer	11	TE$
Bijzonderheden	10	BY$

Met deze gegevens kunnen we de verschillende velden dimensioneren:

```
FIELD #1,3 AS BE$,8 AS VO$,6 AS TU$,18 AS NA$,25 AS
          AD$,7 AS PO$,20 AS PL$,11 AS TE$,10 AS BY$
```

Voor we de verschillende velden gaan dimensioneren moeten we eerst het bestand openen met het volgende commando:

```
OPEN "A:BESTAND1.DAT" AS #1 LEN 108
```

Hierna kunnen we alle gegevens net als bij het telefoonboekprogramma invoeren en nadat we akkoord zijn met alle gegevens, naar de bestandbuffer doorsturen met het volgende commando:

```
PUT #1,R
```

Hierin geeft de variabele R het 'recordnummer' aan, zodat ieder record een eigen 'uniek' nummer heeft.
Voor we de gegevens doorsturen naar de buffer kijken we nog even naar de lengte van bijvoorbeeld het veld 'woonplaats'. Deze heeft een lengte toegewezen gekregen van 20 karakters maar een plaats als 'Deventer' heeft maar acht karakters. Om de woonplaats nu op een vaste plaats in het veld te zetten hebben we de commando's LSET en RSET. We dienen nu wel een extra variabele in te voeren om dit netjes op te lossen. Bij alle variabelen voor de invoer van de gegevens draaien we de letters om, dus de woonplaats heeft bij het invoeren de variabele LP$. Het vastleggen in het veld gaat als volgt:

```
LSET PL$=LP$     evenals
LSET NA$=AN$     enz.
```

Alle ingevoerde gegevens zetten we dus aan de linkerzijde van het veld (zie ook hoofdstuk 4). Achter de gegevens worden in het veld spaties opgenomen (MSX-karaktercode 32), dit laatste is belangrijk om straks de gegevens op het beeldscherm of via de printer te kunnen weergeven.
Het opslaan van een telefoonnummer bijvoorbeeld kunnen we het beste doen met het RSET-commando:

```
RSET TE$=ET$
```

Dit heeft het voordeel dat bij het printen van een overzicht de nummers aan de rechterzijde onder elkaar komen.
Ten slotte sluiten we het bestand met CLOSE #1 en alle gegevens gaan vanuit de buffer naar de diskette.

Een voordeel bij de Random Access File is, dat men in een eenmaal geopend bestand zowel kan wegschrijven als lezen. Dat houdt in dat men zonder speciale commando's nieuwe records kan toevoegen. Daartoe dient men wel te weten hoeveel records inmiddels in het totale bestand zijn opgenomen. Het makkelijkste om dit te onthouden is het totale aantal records weg te schrijven in het eerste record, waardoor de echte gegevens pas beginnen in record 'twee'. Hiertoe kunnen we het eerste record opnieuw dimensioneren:

```
FIELD #1, 3 AS R$,105 AS X$
```

Hierna komt het eerdergenoemde FIELD-commando met de 'echte' indeling van het record. Let erop dat het totale aantal karakters gelijk moet zijn! In ons geval is het makkelijker om niets

aan de indeling van het record te veranderen en op de plaats van BE$ het recordnummer weg te schrijven. Als men veronderstelt dat drie posities te weinig zijn, kan men een ander veld kiezen voor het recordnummer, of ...
MSX Disk BASIC heeft een aantal functies die het manipuleren met getallen in bestanden erg eenvoudig maakt. Het recordnummer is altijd een integer, waardoor het mogelijk wordt om de volgende functie te gebruiken om een integer tot '32767' om te zetten in een string van twee bytes.

```
R$=MKI$(R%)
```

De variabele R% is het hoogste recordnummer dat met MKI$ wordt omgezet naar de string 'R$'. Nu dienen we deze string nog op de juiste plaats in het veld te zetten met het volgende commando:

```
LSET BE$=R$          of in één keer
LSET BE$=MKI$(R%)
PUT #1,1
```

Het lezen van het aantal records uit het bestand gaat met het omgekeerde commando, we laten de gehele procedure zien na het openen van het bestand:

```
GET #1,1             lezen eerste record
R%=CVI (BE$)         omzetten string naar integer
```

Het lezen van de andere records gaat met behulp van de variabele R%.

```
FOR I = 2 TO R%
GET #1,I
PRINT NA$,AD$,PL$
NEXT
```

Hier lezen we record één natuurlijk niet in omdat daar alleen het aantal records in te vinden is. Bij het uitvoeren van dit laatste programma, zullen we zien dat alle spaties achter NA$, AD$ en PL$ ook worden weergegeven op het beeldscherm. Met een hulproutine kunnen we deze snel en eenvoudig verwijderen:

```
S$=NA$ : GOSUB 1000 : NA$=S$

1000 FOR I= LEN(S$) TO 1 STEP-1
1010 IF MID$(S$,I,1) <> CHR$(32) THEN 1030
1020 NEXT
1030 S$= LEFT$(S$,I+1): RETURN
```

Omdat we de verschillende variabelen tijdelijk de naam S$ geven kunnen we steeds dezelfde subroutine gebruiken.
Het sorteren gaat ook makkelijker dan bij een sequentieel bestand, daar het nu niet nodig is om delen van het bestand in te lezen en dan te verschuiven. We kunnen nu het eerste en het tweede record inlezen. Het veld dat we hebben gekozen als sorteercriterium vergelijken we en als het veld van het eerste record achter het tweede record hoort te staan, dan maken we van het tweede record het eerste. Het 'oude' eerste record maken we tweede en gaan we nu vergelijken met het derde record; hoort deze ook voor het record in het geheugen, dan schuiven we die ook één plaats naar voren.

`FOR R%= 2 TO R`	bepalen recordnummer
`GET #1,R%`	lezen record (R)
`EB$=BE$:OV$=VO$ ... ENZ`	omzetten variabelen
`GET #1,R%+1`	lezen volgend record
`(R+1)`	
`IF AN$ < NA$ THEN NEXT`	vergelijken
`REM VERWISSELEN RECORDS`	
`PUT #1,R%`	schrijven record
`(R+1→R)`	
`LSET BE$=EB$:LSET`	omzetten variabelen
`VO$=OV$ ... ENZ`	
`PUT #1,R%+1`	schrijven record
`(R→R+1)`	
`NEXT`	

Voor een echt sorteerprogramma is dit nog niet voldoende, maar het principe blijft hetzelfde.

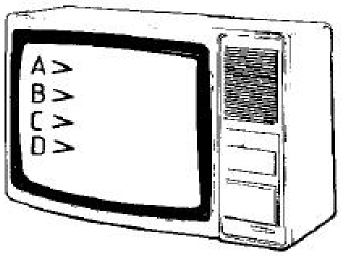
HOOFDSTUK 6

OPSTARTEN VAN MSX-DOS

6.1 Het opstarten

Het opstarten van MSX-DOS kan, zoals reeds in paragraaf 2.3 werd besproken, op twee manieren geschieden en wel door middel van een koude start en door middel van een warme start. We herhalen beide procedures in het kort.
Bij een koude start staan de computer en de randapparatuur (TV, printer, diskdrives) uit.
U zet eerst de randapparatuur aan en vervolgens de computer. Onmiddellijk nadat de computer is aangezet, plaatst u de systeemdiskette in de diskdrive.
Bij een warme start staan de computer en de randapparatuur aan. U plaatst in dit geval de systeemdiskette in de diskdrive en drukt daarna op de reset-knop van uw computer.
Bij meer dan één diskdrive dient de diskette in drive A te worden geplaatst. Bij één diskdrive is dat automatisch diskdrive A, ook wel de 'default drive' genoemd. Het Engelse woord 'default' kunt u het beste vertalen met 'standaard'.
Bij het opstarten met een nieuwe MSX-DOS-systeemdiskette is het mogelijk dat er automatisch een hulpprogramma wordt geladen. Philips heeft daartoe een hulpprogramma gemaakt met de naam DOSHLP.COM. Raadpleeg eerst de gebruiksaanwijzing die zich bij uw systeemdiskette bevindt voor de juiste opstartprocedure.
Heeft u de files MSXDOS.SYS en COMMAND.COM gekopieerd naar een nieuwe diskette, bijvoorbeeld met behulp van het BATCH-programma 'NIEUW.BAT' dat wordt besproken in hoofdstuk 8, dan verschijnen na het aanzetten van de diskdrive en de computer (of na het indrukken van de reset-knop) na enige tijd de boodschappen die we in paragraaf 2.3 hebben leren kennen, op het beeldscherm.

6.2 De cursor



Nadat u de datum heeft ingevoerd en een <RETURN> heeft gegeven, verschijnt de cursor:

```
A>
```

De letter geeft aan vanuit welke diskdrive er is opgestart, altijd diskdrive A. De diskdrive die het laatste is gebruikt, wordt ook wel de default diskdrive genoemd. Het opstarten van MSX-DOS is alleen mogelijk met de systeemdiskette geplaatst in diskdrive A. Bevindt de MSXDOS.SYS-file zich niet op de diskette in diskdrive A, dan zal het systeem uit zichzelf opstarten in de Disk BASIC. Indien dat het geval is, kan men MSX-DOS alleen opstarten door de computer uit en aan te zetten en direct daarna een diskette voorzien van MSX-DOS in diskdrive A te plaatsen (koude start).
Men kan de computer ook in Disk BASIC laten staan en de diskette voorzien van MSX-DOS in diskdrive A plaatsen en daarna de reset-knop op de computer indrukken (warme start). Is er opgestart met MSX-DOS, dan kan men met een speciaal daarvoor bestemd commando naar het Disk BASIC overgaan en met een ander commando weer terug naar het MSX-DOS-besturingssysteem.

De cursor geeft dus aan welke diskdrive is ingeschakeld (default). Om van diskdrive te veranderen, dient men achter de cursor de benaming van de andere diskdrive op te geven die u wilt gaan gebruiken, gevolgd door een dubbele punt. Zoals u heeft kunnen lezen in hoofdstuk 1, kent MSX-DOS maximaal vier diskdrives met de benaming A tot en met D. Overschakelen naar bijvoorbeeld diskdrive B gaat met het volgende commando:

```
A>B:
```

Met dit commando zal na het geven van een RETURN tevens de cursor veranderen in:

```
B>
```

Nu is diskdrive B de default diskdrive. Zolang de diskdrive verder niet wordt aangesproken, zult u buiten deze cursor-verandering niets merken. Pas bij het geven van een commando waarbij de diskdrive wordt aangesproken, zal de computer u erop attenderen dat er een andere diskdrive default is en daar we met één diskdrive werken dient er een andere diskette te worden geplaatst.

Nu blijft diskette B ingesteld staan als de default diskette tot u een andere diskdrive gaat benoemen als default diskdrive.
De volgende cursors zijn bij MSX-DOS dus mogelijk:

```
A>
B>
C>
D>
```

6.3 Default diskdrive en diskette

Na het opstarten is diskdrive A altijd de default drive, het kiezen van een andere diskdrive gaat dus met één van de volgende commando's:

`B:` of `C:` of `D:`

De letters mogen zowel hoofdletters als kleine letters zijn, in de computer worden alle letters van commando's omgezet naar hoofdletters. Dus de commando's

`b:` of `c:` of `d:`

zijn ook correct en zullen voor de computer gelijk zijn aan de commando's in hoofdletters.
Bij het gebruik van één interface heeft u de keuze uit diskdrive A en diskdrive B; met het commando B: gaat u over naar de B-cursor die op het scherm zal verschijnen, verder gebeurt er niets. Zodra u om de directory vraagt of een ander commando invoert, zal de computer het commando uitvoeren met gebruikmaking van diskdrive B. U ziet de LED op diskdrive B dan ook branden ten teken dat hij in werking is. Diskdrive B blijft nu de default diskdrive tot u weer een andere cursor oproept en een commando voor de diskdrive geeft.
Bij het invoeren van een combinatie die niet mogelijk is, geeft de computer de melding:

```
Invalid drive specification
A>■
```

Dus bij het gebruik van één interface zullen de commando's C: of D: resulteren in de voorgaande foutmelding.

Heeft men slechts één diskdrive ter beschikking, dan zal het commando voor een andere diskdrive alleen tot gevolg hebben dat de cursor verandert. Zodra er een ander commando wordt

gegeven, herinnert de computer u eraan dat er eerst een andere diskette moet worden geplaatst.

```
Insert diskette for drive B:
and strike a key when ready ■
```

Na het verwisselen van de diskette drukt u een willekeurige toets in en het eerder ingevoerde commando wordt uitgevoerd. Tevens blijft diskette B nu de default diskette, hetgeen door de cursor wordt aangegeven.
Gaat men met het commando A: weer terug naar diskette A, dan zal de computer na het eerstvolgende commando vragen om diskette A te plaatsen.

6.4 De inhoudsopgave van de diskette

Alle diskettes hebben een gereserveerde ruimte waarin de gegevens zich bevinden van de verschillende files die op de diskette staan (zie ook paragraaf 1.3). Natuurlijk is het voor de gebruiker van belang om snel te kunnen zien welke programma's of files op de diskette staan. Hiertoe kent MSX-DOS het commando:

```
DIR
```

DIR staat voor het Engelse woord 'directory' dat inhoud betekent. Heeft u de computer opgestart met MSX-DOS en het



beeldscherm laat de cursor zien, dan verwacht deze een commando, bijvoorbeeld DIR. Uw scherm ziet er dan als volgt uit:

```
A>dir                    of                A>DIR
```

Beide resulteren na het geven van een <RETURN> in een opsomming van alle programma's, bestanden en files die op de diskette staan, voorzien van de belangrijkste gegevens zoals de lengte in bytes en de datum waarop de file is weggeschreven naar de diskette of voor het laatst is bijgewerkt. Na het opstarten met uw systeemdiskette en het geven van het commando DIR zullen altijd de volgende files in de directory staan:

```
MSXDOS  SYS 2560 1-01-84
COMMAND COM 6528 1-01-84
2 files 352256 bytes free
A>■
```

Afhankelijk van de MSX-DOS-versie kan de lengte in bytes per file verschillen. Deze twee files zijn zoals we weten essentieel voor MSX-DOS en zonder deze kan het systeem niet werken. Het is dus belangrijk om deze files te kopiëren naar al uw diskettes, u heeft dan altijd de beschikking over MSX-DOS onafhankelijk van de diskette die u bij het opstarten gebruikt.
Het DIR-commando wordt in hoofdstuk 7 uitgebreid besproken.

6.5 Formatteren

Het commando FORMAT zorgt ervoor dat de diskette gereed wordt gemaakt om programma's, bestanden en files te ontvangen (zie ook paragraaf 1.1). Alle nieuwe diskettes dient men dan ook voor het gebruik te *formatteren*.
Diskettes die al zijn geformatteerd onder MSX Disk BASIC zijn zonder meer geschikt voor MSX-DOS. U kunt uw diskettes, waarop alleen programma's in Disk BASIC staan, aanvullen met de twee MSX-DOS-files.

Zodra de cursor op het beeldscherm verschijnt, voert u het commando in voor het formatteren:

```
A>FORMAT
```

Na het geven van een <RETURN> verschijnt de volgende vraag op uw beeldscherm:

```
Drive name? (A,B) ■
```

Kies zorgvuldig de juiste diskdrive waarin zich de te formatteren diskette bevindt. Na het intoetsen van de juiste letter A of B, komt de computer voor alle zekerheid nog met de volgende mededeling:

```
Strike a key when ready ■
```

Indien de te formatteren diskette nog niet is geplaatst, kunt u dat alsnog doen.
Heeft u méér dan twee diskdrives, dan zal de vraag luiden:

```
Drive name? (A,B,C,D) ■
```

Na het plaatsen van de te formatteren diskette en het indrukken van de juiste letter voor de diskdrive waarin die diskette is geplaatst, kunt u een willekeurige toets indrukken en zal de diskette worden geformatteerd. MSX-DOS meldt u dat het formatteren is volbracht met de volgende boodschap:

```
Format complete
A>■
```

Het is verstandig om vooraf een aantal diskettes te formatteren. Daarna kunt u de systeemfiles die nodig zijn voor het opstarten van MSX-DOS kopiëren naar de geformatteerde diskettes.
Deze procedure geldt ook bij het gebruik van meer diskdrives. Bij het gebruik van één diskdrive is het wel verstandig de nieuwe diskette eerst in de diskdrive te plaatsen voor u het commando FORMAT geeft. Bij het intoetsen van de letter B na het commando **FORMAT**, zal de computer vragen om 'diskette B'. Na het formatteren komt automatisch de A-cursor terug. Indien we diskette B in de diskdrive laten zitten en een ander commando geven, zal de computer eerst vragen om diskette A.

6.6 Back-up of kopie

Het verdient aanbeveling om back-up-diskettes (kopieën) te maken van al uw diskettes. Indien een diskette is beschadigd of gegevens worden verminkt, zult u blij zijn met de originele diskette (zie ook paragraaf 1.4). De MSX-DOS-files kunt u natuurlijk het beste op al uw diskettes overnemen, zodat opstarten in MSX-DOS met iedere diskette mogelijk is.
Een complete back-up van de systeemdiskette is ook raadzaam, werk daarna alleen met die back-up-diskette en bewaar de originele systeemdiskette apart van uw andere diskettes.
Voor het maken van een kopie kent MSX-DOS het commando:

```
A> COPY
```

Dit commando brengt een file over van de ene diskette naar de andere.
In hoofdstuk 3 kwamen de wildkarakters aan de orde, waarmee het mogelijk is om de systeemdiskette in zijn geheel te kopiëren met het commando:

```
A> COPY *.* B:
```

Alle files van de systeemdiskette worden nu overgebracht naar diskette B, die daarna als werkdiskette fungeert.
Bij het kopiëren zal de computer de eerste file lezen en die in zijn geheugen opslaan. Zodra het geheugen vol is, zal de computer naar diskette B vragen om de inhoud van zijn geheugen weg te kunnen schrijven. Na het wegschrijven vraagt de computer naar diskette A om de rest van de file te kopiëren, of om een volgende file te kopiëren. Dit blijft zich herhalen tot alle files zijn gekopieerd. Ten slotte meldt de computer dat alle files zijn gekopieerd met bijvoorbeeld de volgende mededeling:

```
11 files copied
A>■
```

Het getal geeft aan hoeveel files er in het totaal zijn gekopieerd. Tevens ziet u tijdens het kopiëren de desbetreffende filenaam of filenamen op uw beeldscherm.
U kunt natuurlijk ook alleen de COMMAND.COM- en de MSXDOS.SYS-files overbrengen naar een andere diskette met het COPY-commando, bijvoorbeeld naar een diskette die al is voorzien van uw eigen programmafiles, of naar nieuwe diskettes die u heeft geformatteerd. Als u alleen de twee MSX-DOS-files wilt kopiëren naar een andere diskette, voer dan het volgende commando in:

```
A> COPY MSXDOS.SYS B:
```

Let hierbij goed op de spaties tussen de verschillende commando's. Allereerst vertellen we de computer dat hij moet kopiëren (COPY), na de spatie komt de naam van de te kopiëren file (MSXDOS.SYS), dan weer een spatie en de laatste opdracht bevat de naam van de diskette waar de kopie naartoe moet (B:). Na het geven van een <RETURN> zal de computer de file MSXDOS.SYS inlezen vanaf de default diskette en daarna vragen om diskette B te plaatsen om de file weer weg te schrijven. Is de file weggeschreven, dan zal de computer dat melden met:

```
1 file copied
A>■
```

Nu dienen we de tweede systeemfile nog te kopiëren met het commando:

```
A> COPY COMMAND.COM B:
```

Weer moet men op verzoek van de computer van diskette wisselen en na het kopiëren zal de computer dat melden en hebben we een back-up van de MSX-DOS-files.

Bij het gebruik van twee diskdrives zal de computer niet steeds om diskette B vragen. De computer gaat er dan van uit dat er een diskette is geplaatst in diskdrive B. Mocht u dat per ongeluk zijn vergeten, dan zal de computer dat melden en kunt u alsnog een diskette in diskdrive B plaatsen en het kopiëren vervolgen.
Het COPY-commando kent nog meer mogelijkheden, die bij de specifieke MSX-DOS-commando's worden behandeld in hoofdstuk 7.
Ook het snel kopiëren van dezelfde files naar meer diskettes zonder steeds opnieuw het COPY-commando in te typen, is mogelijk met behulp van de 'editing-toetsen', hierover meer in hoofdstuk 9.

HOOFDSTUK 7

MSX-DOS-COMMANDO'S

7.1 Wat zijn commando's?

Het communiceren tussen de computer en alle aangesloten apparatuur gaat door middel van commando's. Door het invoeren van een MSX-DOS-commando via uw toetsenbord kunt u de computer vragen (opdracht geven) opdrachten uit te voeren. Zoals we in het vorige hoofdstuk hebben kunnen lezen, bestaan er commando's om files te kopiëren, maar ook om ze op het beeldscherm weer te geven, te wissen of om ze een andere naam te geven. Ook zijn er commando's om een programma uit te laten voeren en om de datum en tijd te veranderen. Tevens kunt u langs deze weg het beeldscherm en de printer instellen en heeft u ook de mogelijkheid om meer commando's tot één commandofile samen te voegen.
De commando's om alle files te tonen die op de diskette staan of om een diskette te formatteren zijn inmiddels bekend.
MSX-DOS kent twee soorten commando's: directe en indirecte commando's. Directe commando's zijn opgeslagen in de commando-processor en behoren tot de 'echte' MSX-DOS-commando's. Indirecte commando's zijn namen van 'commandofiles' en van 'commandoprogramma's'. In het Engels noemen ze de directe commando's 'internal' en indirecte 'external' commando's. Het komt erop neer dat MSX-DOS er altijd van uitgaat dat het achter de cursor ingetoetste woord een 'bekend en in de computer opgeslagen' commando is, of de naam van een op de diskette staande file of programma. MSX-DOS kijkt na het invoeren of het commando voldoet aan één van de volgende eisen:

Een MSX-DOS-commando.
Een filenaam voorzien van de extension .COM.
Een filenaam voorzien van de extension .BAT.

De meest eenvoudige en meest gebruikelijke zijn de afzonderlijke MSX-DOS-commando's die niet zichtbaar zijn daar ze, zoals

u in hoofdstuk 2 heeft kunnen lezen, zijn opgenomen in de 'COMMAND.COM'-file en bij het opstarten worden overgebracht naar de commando-processor. Het intoetsen van de commando's gaat na het indrukken van <RETURN> gepaard met een ogenblikkelijke uitvoering van dat commando.

Is het geen 'bekend' MSX-DOS-commando, dan moet het een commando zijn, dat als programmafile op diskette staat. Voordat de commando's worden uitgevoerd, leest de computer ze in vanaf de diskette. Indien de diskette niet is voorzien van de desbetreffende commandofile, zal MSX-DOS deze dus niet kunnen vinden en uitvoeren, hetgeen leidt tot de volgende melding:

```
Bad command or file name
```

Alle files voorzien van de extension .BAT zijn commandofiles, dat wil zeggen programma's die bestaan uit een aantal MSX-DOS-commando's.
Alle commandofiles voorzien van de extension .COM, zijn commandofiles in andere programmeertalen, bijvoorbeeld machinetaal.
Omdat commandofiles met de extension .BAT uit MSX-DOS-commando's bestaan, kunt u die zelf creëren en toevoegen aan de bestaande commando's in de computer. Een commandofile, die u zelf creëert, noemen we een BATCH-programma.

De volgende commando's met een verkorte uitleg zijn beschikbaar onder MSX-DOS:

BASIC	Naar MSX Disk BASIC gaan.
COPY	Kopiëren van gespecificeerde files.
DATE	Weergeven en veranderen van de datum.
DEL	Wissen van gespecificeerde files.
DIR	Laat de files op de diskette zien, eventueel gespecificeerd.
ERASE	Gelijk aan het commando DEL.
FORMAT	Deelt een nieuwe diskette in voor het MSX-formaat.
MODE	Stelt de karakterbreedte van het beeldscherm in.
PAUSE	Pauzeert voor invoer in een BATCH-programma.
REM	Geeft een boodschap weer in een BATCH-programma.
REN	Verandert de filenaam.
RENAME	Gelijk aan het commando REN.
TIME	Weergeven en veranderen van de tijd (optie).
TYPE	Geeft de inhoud van een gespecificeerde file op het beeldscherm.
VERIFY	Schakelt VERIFY in en uit.

7.2 Scheidingstekens

Een MSX-DOS-commando dient gescheiden te zijn van de eventueel daarachter komende getallen of filenamen. Zoals we al hebben gezien bij het weergeven van de inhoudsopgave, is het mogelijk om achter het commando een diskette of diskdrive aan te geven. In dit boek gebruiken we als scheidingstekens altijd de *spatie*, meer spaties zijn ook toegestaan.
Dit is speciaal gedaan om geen verwarring te laten ontstaan met andere scheidingstekens, die bij sommige commando's wel en bij andere weer niet van toepassing zijn. Het is dan ook raadzaam om *alleen* de spatie te gebruiken. Om toch zo volledig mogelijk te zijn, geven we hier de scheidingstekens die bij nagenoeg alle commando's zijn toegestaan. Tussen het opgeven van verschillende filenamen, apparaten e.d. zijn deze scheidingstekens in principe ook toegestaan.

'='	het 'is gelijk'-teken
','	de komma
';'	de puntkomma
' '	de spatie (één of meer)
'T'	de TAB

Ons advies is dus: gebruik altijd spaties om commando's en filenamen van elkaar te scheiden, dit geeft tevens een goed overzicht van het commando op het beeldscherm.
In appendix D geven we een overzicht omtrent het toepassen van de andere scheidingstekens.

7.3 Het gebruik van de commando's

Voor we de specifieke MSX-DOS-commando's gaan behandelen is een waarschuwing op zijn plaats. Regelmatig vraagt de computer om het indrukken van een willekeurige toets. U mag dan iedere toets indrukken en de computer gaat verder met het commando. Er is echter één combinatie (van toetsen) die beslist niet mag worden gebruikt tijdens uitvoering van commando's, dat is <CONTROL>-C (^C). Dit 'controlekarakter' stopt de uitvoering van het commando.

Zoals in de vorige paragraaf al aangegeven dienen de commando's te worden gescheiden door een spatie. Een uitzondering is de benoeming van een andere diskdrive of diskette. Staat een bepaalde file op een andere diskette of bevindt de bewuste diskette zich in een andere diskdrive, dan dient men vóór de filenaam de juiste benaming van die diskdrive of diskette op te nemen. Tussen deze aanduiding en de filenaam mag beslist geen spatie staan; ze moeten aaneengesloten worden ingetypt.

7.4 MSX-DOS-commando's

BASIC *Dit commando dient voor het verlaten van MSX-DOS en over te gaan naar MSX Disk BASIC.*
Indien men achter het commando BASIC een naam opneemt van een BASIC-programma, zal dat BASIC-programma automatisch worden geladen en geRUNd nadat Disk BASIC is geactiveerd.

Voorbeeld: `BASIC TESTPROG.BAS`

Nadat BASIC is geactiveerd, zal het programma TESTPROG.BAS automatisch worden geladen en geRUNd. De programmanaam moet volledig worden opgegeven inclusief de extension; het gebruik van 'wildkarakters' is bij dit commando niet toegestaan.
Denk eraan dat met dit commando de geheugenindeling van de computer verandert, daar die indeling voor MSX Disk BASIC en MSX-DOS verschillend is.
Om terug te keren naar MSX-DOS dient u het statement 'CALL SYSTEM' te gebruiken, of de afgekorte versie '– SYSTEM'.

COPY *Dit commando dient voor het kopiëren van één of meer files van de ene naar de andere diskette. Dit commando werkt exact gelijk voor zowel één diskdrive als twee diskdrives.*

Voorbeeld: `COPY TESTPROG.BAS TEST.BAS`

Dit commando zal het programma TESTPROG.BAS kopiëren naar dezelfde diskette maar onder een andere naam, namelijk TEST.BAS. Een programma of file kopiëren op dezelfde diskette onder dezelfde naam is mogelijk, doch moet in de praktijk worden vermeden, daar bij lange files de informatie gedeeltelijk verloren kan gaan. Na het kopiëren van één programma, bestand of file krijgt u de melding:

```
1 file copied
```

De tweede naam achter de naam van het te kopiëren programma is een optie en kan worden vervangen door een letter om daarmee een andere diskette of diskdrive aan te geven, waardoor het programma onder dezelfde naam naar de opgegeven diskette wordt gekopieerd.

```
COPY TESTPROG.BAS B:
```

zal dus het programma TESTPROG.BAS kopiëren van de default diskette naar diskette B:. Andere combinaties zijn natuur-

lijk ook mogelijk, om bijvoorbeeld vanaf een andere diskette naar de default diskette te kopiëren, geldt het volgende commando:

```
COPY C:TESTPROG.BAS
```

Het programma TESTPROG.BAS wordt nu vanaf diskette C gekopieerd naar de default diskette. Indien men van een bepaalde diskette naar een andere bepaalde diskette wil kopiëren, dient men natuurlijk beide diskdrives op te geven.

```
COPY C:TESTPROG.BAS D:
```
of
```
COPY C:TESTPROG.BAS D:TEST.BAS
```

In het eerste geval wordt het programma gekopieerd en onder dezelfde naam weggeschreven naar diskette D. In het tweede geval zal het programma ook naar diskette D worden gekopieerd, maar het krijgt nu de nieuwe naam TEST.BAS.

Tijdens kopiëren is het ook mogelijk verschillende files aan elkaar te koppelen tot één file. De te kopiëren files worden opgegeven, gescheiden door een '+'-teken.

```
COPY A:TEST1.TXT + B:TEST2.TXT A:TOTAAL.TXT
```

De files TEST1.TXT van diskette A en file TEST2.TXT van diskette B worden samengevoegd tot één nieuwe file en weggeschreven naar diskette A onder de naam TOTAAL.TXT.
Zodoende zijn bijvoorbeeld verschillende tekstfiles samen te voegen tot één grote tekstfile. In het genoemde voorbeeld zal TEST2.TXT achter TEST1.TXT worden opgenomen in de nieuwe file.
Normaal gesproken worden alleen ASCII-files aan elkaar gekoppeld, over het algemeen tekstfiles gemaakt met een tekstverwerkend programma. Ook BASIC-programma's kunnen als tekstfiles worden opgeslagen, ze moeten dan worden geSAVEd met de A-optie. Het einde van een ASCII-file wordt altijd gekenmerkt door een '<CONTROL>-Z' (^Z) code; dat is de MSX-karaktercode 26.
Binaire files en programma's hebben geen ^Z-code aan het einde. De lengte van een dergelijke file moet voor het kopiëren uit de inhoudsopgave worden gelezen. Om dat aan te geven gebruikt men de '/B'-switch, bijvoorbeeld:

```
COPY/B TESTA.COM + TESTB.COM
```

De lengtes van de files TESTA.COM en TESTB.COM worden

nu uit de inhoudsopgave gelezen. In dit voorbeeld wordt de file TESTB.COM achter TESTA.COM geplaatst en heet de nieuwe file TESTA.COM. De originele file TESTA.COM bestaat dus niet meer en is vervangen door een samenvoeging van beide files.

Om een ^Z-code toe te voegen aan een nieuwe file wordt de '/A'-switch gebruikt, bijvoorbeeld:

```
COPY TESTA.COM/B TESTB.COM/A
```

De lengte van de file TESTA.COM wordt uit de inhoudsopgave gelezen. Na het kopiëren wordt er een ^Z-code toegevoegd aan de nieuwe file TESTB.COM.
De '/A'-switch zal één ^Z-code toevoegen aan de nieuwe file. Het toevoegen van een tweede ^Z-code aan een file kan alleen bij tekstfiles zoals het volgende voorbeeld laat zien:

```
COPY TEST1.ASC/B TEST2.ASC/A
```

Nu wordt de lengte van de ASCII-file TEST1.ASC uit de inhoudsopgave gelezen; dat betekent inclusief de normale ^Z-code die al als eindefile-teken was opgenomen. Na het kopiëren wordt er een ^Z-code toegevoegd aan de nieuwe file TEST2.ASC. Deze nieuwe file heeft nu aan het einde twee ^Z-codes: de eerste van de oorspronkelijke file TEST1.ASC, de tweede door de '/A'-switch.
Een switch (/B of /A) blijft net zolang actief totdat de computer een nieuwe switch tegenkomt.

```
COPY TESTA.COM/B + TESTB.COM TESTC.COM/A
```

Nu wordt de lengte van de files TESTA.COM en TESTB.COM uit de inhoudsopgave gelezen en wordt er een ^Z-code toegevoegd aan het einde van de nieuwe file TESTC.COM.

Bij het kopiëren is ook het gebruik van wildkarakters toegestaan, bijvoorbeeld:

```
COPY *.TXT COMBI.TEX
```

Alle files voorzien van de extension .TXT worden eerst samengevoegd en dan gekopieerd tot een nieuwe file COMBI.TEX. De files die worden gekopieerd zullen natuurlijk ook als afzonderlijke files op de diskette blijven staan.

```
COPY *.TXT + *.TEX COMBI.TTT
```

Alle files voorzien van de extension .TXT worden gekopieerd en samengevoegd met alle files voorzien van de extension .TEX en weggeschreven als een nieuwe file COMBI.TTT op dezelfde diskette.

```
COPY *.TXT ALLES.TXT
```

Dit commando geeft aanleiding tot het wissen van een file indien de file ALLES.TXT al bestaat. Om de files met de extension .TXT te kopiëren kijkt de computer in de inhoudsopgave naar de eerste file met die extension en leest hem dan in, daarna de volgende file enz. Gelijktijdig wordt er in het geheugen een nieuwe file gemaakt (ALLES.TXT). Indien alle files zijn ingelezen, dus ook de reeds bestaande file ALLES.TXT, wordt de nieuwe file weggeschreven onder de naam ALLES.TXT. Omdat die file al bestaat, wordt deze file over de oude heen geschreven en zijn we die kwijt; hij maakt natuurlijk wel deel uit van de nieuwe file.

```
COPY ALLES.TXT + *.TXT
```

Dit commando kan fouten veroorzaken, daar alle files met de extension .TXT worden ingelezen net als bij het vorige voorbeeld, doch voordat alle files met de extension .TXT worden gelezen, leest de computer eerst de file ALLES.TXT. Na het lezen van ALLES.TXT zal de computer vooraan in de directory beginnen met het zoeken naar files eindigend op .TXT. Eén voor één zullen de files achter ALLES.TXT in het geheugen worden opgenomen. Op een gegeven moment komt de computer de reeds bestaande file ALLES.TXT ook tegen en leest die dan ook in (voor de tweede keer). Na het inlezen van alle files wordt de nieuwe file 'ALLES.TXT' naar de diskette geschreven. Het komt er dus op neer dat de 'oude' file ALLES.TXT twee keer voorkomt in de nieuwe file.

Waarschuwing:
De twee hiervoor genoemde COPY-commando's mogen dan wel erg gemakkelijk lijken om stukken tekst samen te voegen tot één lange tekstfile, maar het gebruik hiervan moet toch worden ontraden. Het beste kan men de tekstfiles samenvoegen en wegschrijven onder een nog niet bestaande naam en een andere extension. Na het kopiëren kan men met het 'RENAME'-commando de nieuwe tekstfile iedere gewenste naam geven.

```
COPY TEKST?.TXT TEKST?.TTT
```

Met dit commando worden alle files die beginnen met 'TEKST' met daarachter nog één karakter en eindigend op .TXT geko-

pieerd naar nieuwe files met dezelfde naam, maar met een andere extension. De volgende files zullen bijvoorbeeld worden gekopieerd:

```
TEKST1.TXT      wordt     TEKST1.TTT
TEKST3.TXT      wordt     TEKST3.TTT
TEKST  .TXT     wordt     TEKST  .TTT
```

Gebruik bij het kopiëren nooit één of meer vraagtekens aan het einde van een filenaam, het volgende commando zal meer files kopiëren dan men verwacht.

```
COPY TEST??.BAS TEST??.TXT
```

Alle files die beginnen met 'TEST' met daarachter nog één of twee karakters zullen worden gekopieerd. Dus niet alléén files die bestaan uit zes karakters waarvan de eerste vier bestaan uit 'TEST'. De volgende files worden bijvoorbeeld gekopieerd:

```
TEST12.BAS      wordt     TEST12.TXT
TESTAA.BAS      wordt     TESTAA.TXT
maar ook
TEST3.BAS       wordt     TEST3.TXT
TESTB.BAS       wordt     TESTB.TXT
TEST  .BAS      wordt     TEST  .TXT
```

Daar alle filenamen worden opgeslagen in de vorm van acht karakters, zal de naam TEST3.BAS opgeslagen zijn als 'TEST3   .BAS'. Tussen de drie en de punt bevinden zich (voor de computer) nog drie spaties. Een spatie voldoet aan de kenmerken van een willekeurig karakter. Voor het vraagteken als 'wildkarakter' maakt het niets uit of er een spatie of een ander karakter op die bewuste plaats staat.

COPY *Dit commando kan ook worden gebruikt om files te kopiëren naar andere apparatuur.*

Voorbeeld: `COPY TEKST.TXT CON`

Dit commando zal de inhoud van de file TEKST.TXT kopiëren naar CON (het CONsole, beeldscherm). U krijgt op het beeldscherm de tekst te zien waaruit deze file bestaat. In plaats van CON kunt u ook andere apparatuur invullen (AUX) en natuurlijk de printer, bijvoorbeeld:

```
COPY TEKST.TXT LST
```

zal de tekstfile naar de printer sturen en alle informatie op papier zetten. In plaats van de aanduiding 'LST' kan men ook 'PRN' gebruiken.

Denk eraan dat alleen ASCII-files zichtbaar kunnen worden gemaakt.

```
COPY CON TEST.TES
```

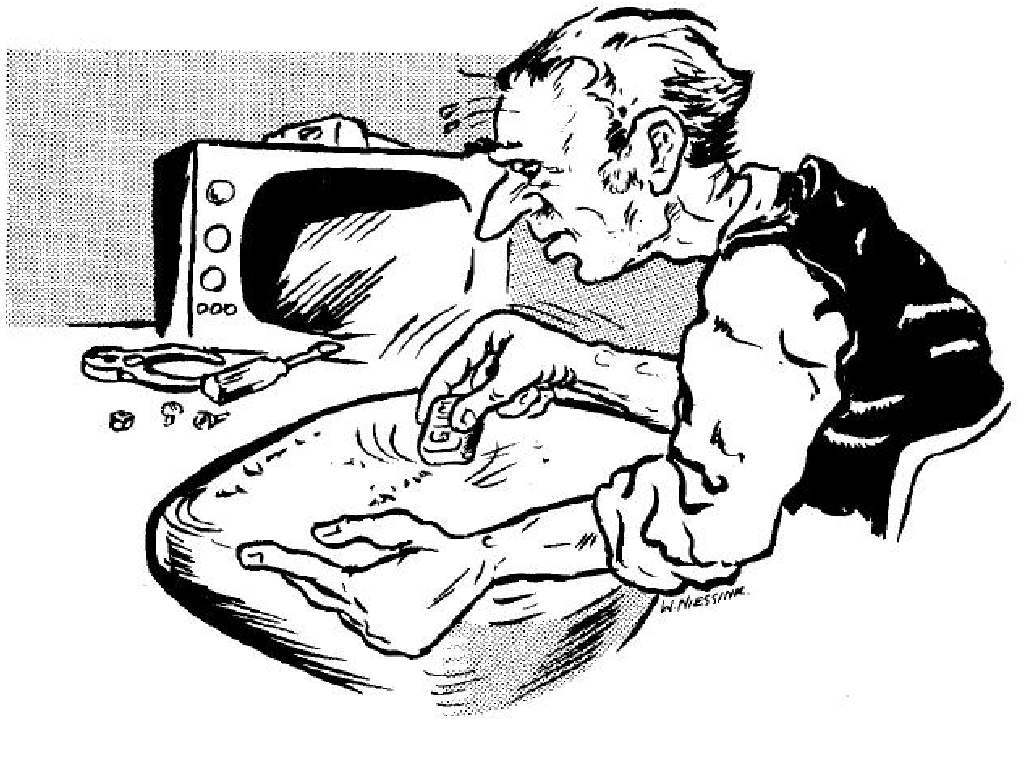
Dit commando zal alles dat u via het toetsenbord invoert, overbrengen naar de default diskette onder de naam TEST.TES. Dit gebeurt pas na het afsluiten van de tekst met een ^Z-code daar dit alleen mogelijk is als ASCII-file. Meer omtrent het overbrengen van teksten en commandofiles naar de diskette vindt u in het hoofdstuk over 'BATCH'-programma's.

DATE *Dit commando dient voor het controleren en wijzigen van de datum. De computer zal de laatst ingevoerde datum opnemen in de inhoudsopgave bij iedere nieuwe file die u op diskette zet.*

Voorbeeld: `DATE <M> - <D> - <Y>`

In andere landen kan de volgorde iets anders zijn. De Amerikaanse versie gebruikt de volgorde <Y>-<M>-<D>.

Nadat u het commando DATE heeft ingetypt kunt u direct de datum invoeren. Voert u na het DATE-commando geen datum in maar drukt u eerst op de <RETURN>-toets, dan zal de volgende mededeling op uw beeldscherm verschijnen:



```
Current date is <dag> <M>-<D>-<Y>
Enter new date: ■
```

In de eerste regel staat de bij het opstarten ingevoerde datum. Heeft u bij het opstarten van MSX-DOS geen datum ingevoerd, dan ziet u het volgende op het beeldscherm:

```
Current date is Sun 1-01-1984
Enter new date: ■
```

Wilt u de op uw beeldscherm weergegeven datum niet veranderen, dan de <RETURN>-toets indrukken en de cursor verschijnt weer.
De nieuwe datum dient u in te voeren met cijfers, waarbij u de volgende mogelijkheden heeft:

<M> maand	1-12 of 01-12
<D> dag	1-31 of 01-31
<Y> jaar	0-79 of 00-79
	of 80-90
	of 1980-2099

De ingevoerde cijfers moeten van elkaar zijn gescheiden door één van de volgende scheidingstekens:

'-'	koppelstreepje
'.'	punt
'/'	schuine streep

11-28-1990 wordt donderdag 28 november 1990 (THU 11-28-1990)
4/19/85 wordt vrijdag 19 april 1985 (FRI 4-19-1985)
06.20.72 wordt maandag 20 juni 2072 (MON 6-20-2072)

Het invoeren van een foutieve datum leidt tot de volgende mededeling:

```
Invalid date
Enter new date: ■
```

DEL *Dit commando dient voor het wissen (Engels DELete) van één of meer files van de diskette. Achter het commando wordt de te wissen file gespecificeerd.*

Voorbeeld: `DEL TESTFILE.TXT`

De file **TESTFILE.TXT** wordt van de diskette gewist. Zolang er

geen drive wordt opgegeven, zal dat altijd van de default diskette zijn. Door voor de filenaam de diskette of diskdrive in te toetsen, wordt de file van de genoemde diskette gewist, bijvoorbeeld:

```
DEL B:TESTFILE.TXT
```

Het gebruik van wildkarakters is hier toegestaan, hoewel men daar erg voorzichtig mee moet zijn.

```
DEL B:*.TXT
```

Deze opdracht zal alle files met de extension .TXT die zich op diskette B bevinden, wissen. Om alle files ongeacht hun naam of extension van een diskette te wissen, kunnen we dus het volgende commando gebruiken:

```
DEL B:*.*
```

Om vergissing te voorkomen, vraagt de computer voor alle zekerheid of deze 'totale' wisopdracht inderdaad moet worden uitgevoerd:

```
Are you sure (Y/N)? ■
```

Het intoetsen van 'n' of 'N', zal deze opdracht te niet doen, het intoetsen van 'y', 'Y' of alleen <RETURN> zal alle files op diskette B wissen.
In plaats van het commando DEL mag men ook ERASE gebruiken.

DIR *Dit commando toont u de DIRectory (inhoudsopgave) van de diskette. U heeft de keuze uit een totaaloverzicht inclusief de filegegevens, of een verkorte inhoudsopgave die alleen de filenamen laat zien.*

Voorbeeld: `DIR`

U krijgt nu alle filenamen op uw beeldscherm te zien, voorzien van hun extension en de lengte in bytes, de datum waarop ze naar diskette zijn weggeschreven en, indien uw computer is voorzien van een ingebouwde klok, ook de tijd.
Al heeft uw computer geen klok, dan is het toch mogelijk dat er bij verschillende files achter de datum een tijd wordt weergegeven. Deze files zijn dan op een diskette gezet met behulp van een computer die wél is voorzien van een ingebouwde klok.

```
DIR TESTFILE.TXT
```

toont alleen de gegevens van de file **TESTFILE.TXT** op uw beeldscherm.
Het commando DIR komt overeen met de commando's DIR * en DIR *.*, die ook zijn toegestaan.

```
DIR <filenaam>
```

Toont alle files met dezelfde naam onafhankelijk van de extension en komt dus overeen met het commando DIR <filenaam>.*.

```
DIR .<extension>
```

Dit commando toont alle files met dezelfde (opgegeven) extension en komt overeen met het commando DIR *.<extension>.
Wilt u de inhoudsopgave van een andere dan de default diskette, dan dient u achter het commando de gewenste diskette of diskdrive op te geven:

```
DIR B:
```

Nu wordt de inhoudsopgave van diskette B op het beeldscherm weergegeven. Na het weergeven van deze inhoudsopgave zal de default cursor weer terugkeren. Ook kunt u de 'B:' direct voor een filenaam opgeven (dus zonder spatie!) om die file vanaf diskette B te lezen.
Naast deze mogelijkheden kunt u ook een verkorte versie van de inhoudsopgave opvragen met het volgende commando:

```
DIR /W
```

Dit commando toont alleen de filenaam met zijn extension, ze worden op het beeldscherm in twee kolommen weergegeven (alleen bij MODE 29 of hoger).

Staan er meer dan 23 files op uw diskette, dan zal de eerste file van het beeldscherm verdwijnen voor de laatste file is weergegeven. Om dit te voorkomen kent het DIR-commando de 'paginamode':

```
DIR /P
```

Dit commando toont maximaal 23 files op het beeldscherm. Zijn er meer files op de diskette aanwezig, dan krijgt u de volgende mededeling:

```
Strike a key when ready . . . ■
```

Na het indrukken van een willekeurige toets zal de computer het vervolg van de inhoudsopgave lezen en op uw beeldscherm weergeven.

ERASE *Dit commando is gelijk aan DEL en wist informatie van de diskette. Zie voor verdere bijzonderheden het commando DEL.*

FORMAT *Dit commando maakt de diskette gereed voor het ontvangen en opslaan van MSX-DOS-files.*

Voorbeeld: `FORMAT`

Na dit commando zal de computer de volgende vraag stellen:

```
Drive name? (A,B) ■
```

Heeft u twee diskdrive-interfaces op uw computer aangesloten, dan zal de vraag luiden:

```
Drive name? (A,B,C,D,) ■
```

Na het opgeven van de gewenste diskette of diskdrive zal de volgende boodschap verschijnen:

```
Strike a key when ready . . . ■
```

Nadat u de juiste diskette in de juiste diskdrive heeft geplaatst, kunt u een willekeurige toets indrukken en zal de diskette worden geformatteerd. Zodra het formatteren gereed is, meldt de computer dit met de volgende boodschap:

```
Format complete
A>■
```

MODE *Hiermee kunt u de karakterbreedte van het beeldscherm instellen.*

Voorbeeld: `MODE 40`

Met dit commando zet u het beeldscherm op een breedte van 40 karakters. U kunt iedere breedte opgeven tussen 1 en 40. Afhankelijk van de karakterbreedte wordt het beeldscherm ingesteld op de tekstmode of de grafische mode. Deze schermmode wordt bepaald door het aantal karakters per regel. De grafische mode verkrijgt u door een karakterbreedte van 32 of minder. Een karakterbreedte van 33 tot 40 zet het beeldscherm in de tekstmode.



In de grafische mode kunt u de grafische karakterset van uw toetsenbord goed en volledig weergeven op het beeldscherm. In de tekstmode zullen de grafische karakters niet in hun geheel worden weergegeven.
De tekstmode is natuurlijk het meest geschikt voor het weergeven van teksten en programma's. Uw MSX-computer heeft een standaard-instelling, namèlijk:

```
MODE 37
```

Dit houdt in: de tekstmode met een karakter- (of kolom-)breedte van 37 karakters.

REN
RENAME
Verandert de filenaam (Engels REName) van een te specificeren file in een nieuwe gespecificeerde filenaam.

Voorbeeld:	`REN TEKST.TXT TEKST1.TXT`
of	`RENAME TEKST.TXT TEKST1.TXT`

De bestaande file TEKST.TXT op de default diskette krijgt de nieuwe naam TEKST1.TXT. De inhoud van de file TEKST.TXT zelf blijft gewoon bestaan, alleen in de inhoudsopgave wordt de naam veranderd. Staat de file die van naam moet veranderen niet op de default diskette, dan dient men voor de eerste filenaam de juiste diskette of diskdrive op te geven, zonder spatie:

```
REN B:TEKST.TXT TEKST1.TXT
```

De file op diskette B krijgt nu een andere naam. Het opgeven van een diskdrivenaam voor de tweede filenaam zal worden genegeerd; de file wordt *altijd* op de default diskette van naam veranderd, tenzij voor de eerste filenaam een andere diskette of diskdrive is gespecificeerd.
Het is toegestaan om wildkarakters te gebruiken bij het veranderen van filenamen.

```
REN *.TXT *.TST
REN ABCDE B??B?
```

In het eerste geval zullen alle filenamen met de extension .TXT worden vervangen door dezelfde filenamen, maar nu met de extension .TST. Bij het tweede commando zal de file ABCDE veranderen in BBCBE. Op de plaats van het wildkarakter (?) worden dezelfde letters gehandhaafd. Op de plaatsen waar een letter wordt gegeven, wordt de oude letter vervangen door de nieuw opgegeven letter. Let erop dat in de inhoudsopgave de filenamen *altijd* uit acht karakters bestaan; de naam wordt met spaties aangevuld tot acht karakters. Gebruik daarom nooit één of meer vraagtekens als wildkarakters aan het einde van een filenaam.
Geeft men voor de nieuwe file een filenaam op die overeenkomt met een reeds op de diskette aanwezige file, dan zal er een foutmelding op het beeldscherm verschijnen:

```
Rename error
```

Dezelfde foutmelding krijgen we als we een niet op de diskette staande file van naam willen veranderen.

CALL SYSTEM *Specifiek MSX-DOS-commando dat alleen werkt bij MSX Disk BASIC.*

Voorbeeld:	CALL SYSTEM
of	_SYSTEM

Als men vanuit MSX-DOS met het commando 'BASIC' is overgegaan naar Disk BASIC, kan men zonder de computer uit te zetten of te 'resetten' met één van deze twee genoemde commando's terugkeren naar het MSX-DOS-besturingssysteem. De systeemdiskette voorzien van de files MSXDOS.SYS en COMMAND.COM dient eerst te worden geplaatst in diskdrive A. Deze commando's werken niet als er is opgestart in BASIC of Disk BASIC.

TIME *Dit commando dient voor het controleren en wijzigen van de tijd. Het systeem zal de tijd opnemen in de inhoudsopgave bij iedere nieuwe file die u op diskette zet.*

Voorbeeld: TIME <H><:M><:S>

Na het TIME-commando kunt u direct de nieuwe tijd invoeren. Voert u geen tijd in, dan zal de computer de juiste tijd weergeven (indien die eerder werd ingevoerd) en vragen om een nieuwe tijd:

```
Current time is: <H>:<M>:<S>.<C>
Enter new time: ■
```

Wenst u de tijd niet te veranderen, dan geeft u een <RETURN> en de aanwezige 'current' tijd blijft gehandhaafd. De tijd wordt uitsluitend met cijfers ingevoerd. Tussen de uren, minuten en de eventuele seconden dient een dubbele punt (':') te staan.

H 00-24	zijn uren
M 00-59	zijn minuten
S 00-59	zijn seconden
C 00-99	zijn honderdsten van seconden



De seconden en honderdsten van een seconde zijn een optie en mogen worden weggelaten. Tussen de seconden en honderdsten van een seconde mag geen dubbele punt staan, maar moet men een enkele punt ('.') gebruiken.
MSX-DOS zal de ingevoerde tijd als de nieuwe tijd opnemen indien de ingevoerde cijfers en scheidingstekens correct zijn. Is dat niet het geval, dan krijgt u de volgende boodschap op het beeldscherm:

```
Invalid time
Enter new time: ■
```

en blijft de aanwezige tijd behouden tot u de juiste tijd invoert of een <RETURN> geeft.

Indien uw computer niet is voorzien van een ingebouwde klok zal dit commando niet functioneren en een foutmelding tot gevolg hebben.

TYPE *Geeft de inhoud van een gespecificeerde file weer op het beeldscherm.*

Voorbeeld: `TYPE TEKSTFILE.TXT`

De inhoud van de file TEKSTFILE.TXT wordt nu op het beeldscherm weergegeven. Met het commando DIR kunt u eventueel eerst de juiste filenaam opzoeken.
Eventuele tabsprongen worden omgezet in spaties tot de eerste standaard tabkolom van het beeldscherm. Deze beeldschermtabs staan om de acht kolommen.
Het weergeven van een binaire file zal gepaard gaan met het weergeven van alle eventuele controlekarakters uit die binaire file en zijn onleesbaar op het beeldscherm. Dit commando komt overeen met het commando COPY TESTFILE.TXT CON.

VERIFY *Met dit commando kan alle data die naar de diskette wordt weggeschreven, worden gecontroleerd.*

Voorbeeld: `VERIFY ON`

Nu wordt de verify-mode (controleren) ingeschakeld. Alle data die naar de diskette wordt gestuurd, zal nadat deze op de diskette staat, weer worden gelezen en gecontroleerd aan de hand van de inhoud van de computer. Indien er een fout is opgetreden bij het wegschrijven naar de diskette, verschijnt er een foutmelding op het beeldscherm:

```
DISK I/O error
```

Het uitzetten van de verify-mode gaat met het commando:

```
VERIFY OFF
```

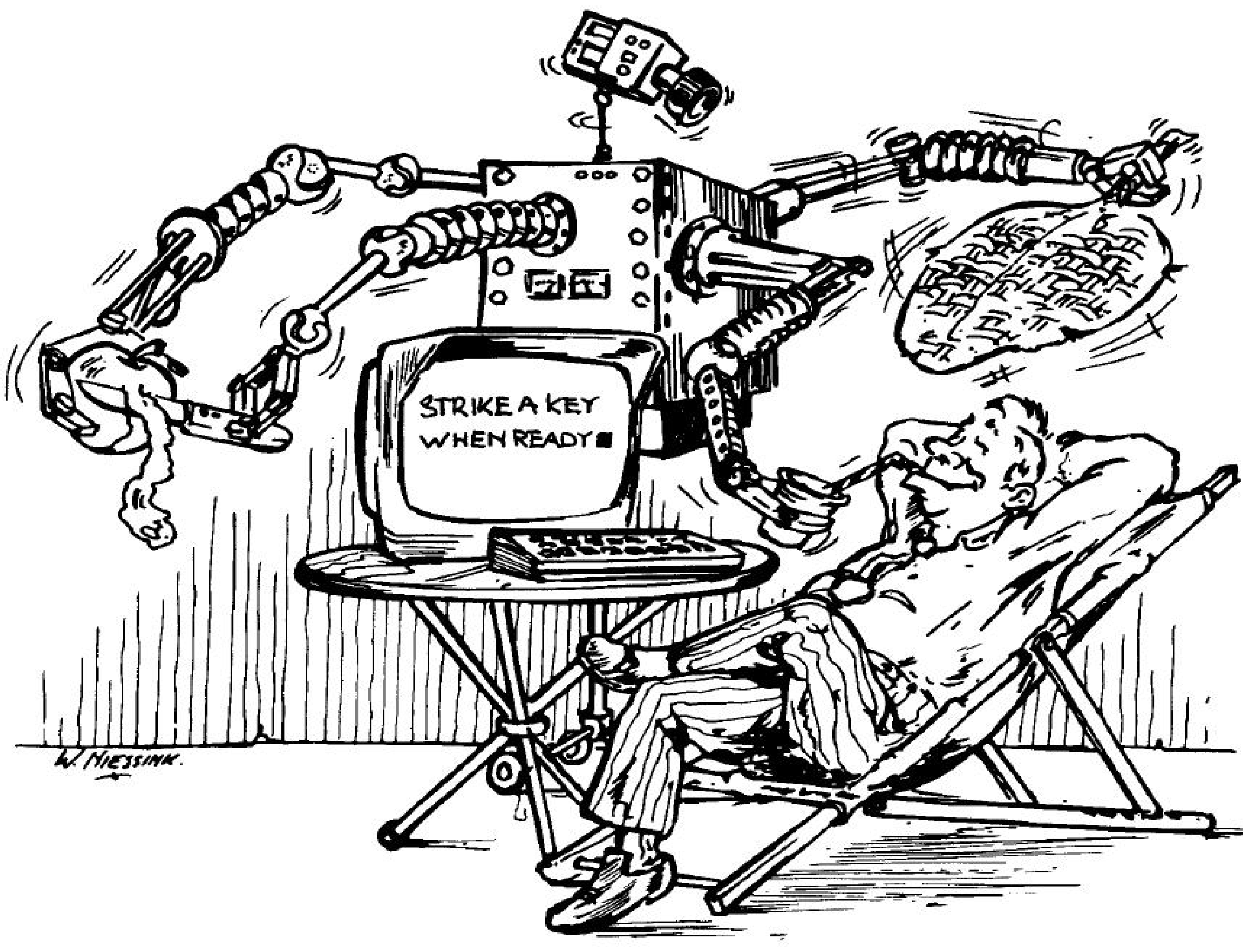
Na het opstarten van MSX-DOS staat de verify-mode altijd op OFF.

7.5 MSX-DOS-commando's voor BATCH-programma's

Alle hiervoor genoemde MSX-DOS-commando's kunnen worden gebruikt in een BATCH-programma (zie ook hoofdstuk 8). Daarnaast heeft MSX-DOS nog twee speciale commando's die alléén in BATCH-programma's kunnen worden toegepast.

PAUSE *Een commando dat kan worden gebruikt om een pauze in te lassen, ten behoeve van bijvoorbeeld het verwisselen van een diskette of het onderbreken van het BATCH-programma op dat punt.*

Voorbeeld: `PAUSE Plaats diskette B`



Na het commando PAUSE kan men een boodschap opnemen, die dan tijdens het uitvoeren van het BATCH-programma wordt getoond. Als scheidingstekens kunt u tussen PAUSE en de mededeling gebruik maken van de spatie, tab-sprong of komma. Na ieder PAUSE-commando verschijnt tijdens het uitvoeren van het BATCH-programma eerst de opgenomen mededeling en daarna verschijnt automatisch de volgende regel op het beeldscherm:

```
Strike a key when ready . . . ■
```

Iedere toets kan worden gebruikt om het BATCH-programma verder te laten gaan, uitgezonderd <CONTROL>-C.
Indien u <CONTROL>-C indrukt, verschijnt de volgende vraag op het beeldscherm:

```
Terminate batch file (Y/N)? ■
```

Als u nu 'Y' invoert, zal het BATCH-programma niet verder worden uitgevoerd en komt de computer weer terug in MSX-DOS. U kunt het PAUSE-commando dus ook gebruiken om een BATCH-programma op bepaalde punten vroegtijdig af te breken.

REM *Het commando REM (REMark) maakt het mogelijk om commentaar en opmerkingen op te nemen die worden weergegeven op het beeldscherm tijdens het uitvoeren van het BATCH-programma.*

Voorbeeld: `REM Deze file kopieert automatisch MSX-DOS`

Als u een BATCH-programma oproept, zal de mededeling achter REM als eerste op het beeldscherm verschijnen (mits ook de eerste regel in het BATCH-programma). Zo kunt u op het beeldscherm laten weergeven wat het BATCH-programma gaat doen. Neemt u hierna het PAUSE-commando op, dan heeft u de gelegenheid om het BATCH-programma af te breken (zie het PAUSE-commando) als uit de inleidende mededeling blijkt dat het niet het juiste BATCH-programma is.

HOOFDSTUK 8

BATCH-PROGRAMMA'S

BATCH-programma's bestaan uit normale MSX-DOS-commando's. De naam van een BATCH-programma, zonder zijn extension, is een nieuw commando dat is samengesteld uit de commando's die in het BATCH-programma zijn opgenomen. Bij MSX-DOS moeten we voor BATCH-programma's de extension .BAT gebruiken.

Zo'n zelf gecreëerd nieuw commando kan bijvoorbeeld 'NIEUW' zijn. Op de diskette zien we dat BATCH-programma staan als:

```
NIEUW.BAT
```

Dit BATCH-programma (als voorbeeld) formatteert en schrijft daarna de twee MSX-DOS-systeemfiles voor het besturingssysteem naar een nieuwe diskette.
Werkt u hoofdzakelijk met Disk BASIC, dan kunt u een BATCH-programma maken dat ervoor zorgt dat de computer na het opstarten direct overgaat naar BASIC en daarna bijvoorbeeld automatisch een keuzemenu weergeeft.

8.1 Wat is een BATCH-programma?

Een BATCH-programma bevat een aantal MSX-DOS-commando's, die samen een bepaalde functie vervullen. Een dergelijke functie vervangt het invoeren van afzonderlijke commando's, die zonder BATCH-programma nodig zouden zijn. Door het intypen van de programmanaam (zonder extension) zal het besturingssysteem het programma vanaf diskette lezen en de afzonderlijke commando's in dat programma uitvoeren, net alsof u ze via het toetsenbord intypt.
Denk eraan dat het BATCH-programma op de diskette voorzien moet zijn van de extension .BAT.

U kunt een BATCH-programma samenstellen met behulp van een tekstverwerkend programma of met het COPY-commando. Een BATCH-programma kan zeer waardevol zijn. Kijk maar eens naar alle commando's die nodig zijn om een nieuwe diskette gereed te maken voor MSX-DOS en daarna te voorzien van de twee MSX-DOS-files.
Als we de daarvoor vereiste commando's onder elkaar zetten, krijgen we het volgende:

FORMAT	<Het systeem vraagt A of B>
COPY A:COMMAND.COM B:	<Kopiëren MSX-DOS-commandofile>
COPY A:MSXDOS.SYS B:	<Kopiëren MSX-DOS-systeemfile>
DIR B:	<Controleren inhoudsopgave van de nieuwe diskette>

Na het intypen en uitvoeren van al deze commando's is de diskette gereed voor gebruik. Het is natuurlijk veel gemakkelijker om deze commando's samen te voegen tot een BATCH-programma dat deze commando's voor ons uitvoert.
In het vorige hoofdstuk zijn de twee speciale MSX-DOS-commando's voor BATCH-programma's behandeld, deze passen we nu direct toe. De inhoud van een dergelijk BATCH-programma kan er als volgt uitzien:

```
REM Dit programma formatteert een
REM nieuwe disk en voorziet
REM hem van de MSX-DOS-files
PAUSE
FORMAT
PAUSE MSX-DOS overzetten?
COPY A:COMMAND.COM B:
COPY A:MSXDOS.SYS B:
DIR B:
```

Indien we dit BATCH-programma op onze (back-up) systeemdiskette zetten onder de naam NIEUW.BAT, dan kunnen we bij het gereedmaken van een nieuwe diskette deze in diskdrive A plaatsen en het programma oproepen. Tijdens het uitvoeren van het BATCH-programma zal de computer ons melden wanneer de nieuwe diskette (B) moet worden geplaatst c.q. verwijderd.
Hebben we twee diskdrives op onze computer aangesloten, dan

plaatsen we de nieuwe diskette natuurlijk in diskdrive B en is het niet nodig steeds van diskette te wisselen.
We gaan in dit voorbeeld uit van één diskdrive.
Na het intypen van het commando 'NIEUW', krijgen we de eerste vier regels op het beeldscherm en daaronder een extra regel als gevolg van het PAUSE-commando:

```
A>NIEUW
A>rem Dit programma formatteert een
A>rem nieuwe disk en voorziet
A>rem hem van de MSX-DOS-files
A>pause
Strike a key when ready . . . ■
```

Na het indrukken van een willekeurige toets leest de computer het volgende commando en verschijnt er op het beeldscherm:

```
A>format
Drive name? (A,B) ■
```

Na het intoetsen van de letter B gaat de computer ogenblikkelijk verder:

```
Strike a key when ready ■
Insert diskette for drive B:
and strike a key when ready■
```

Nu plaatsen we diskette B en drukken een willekeurige toets in. Na het formatteren komt de volgende mededeling op het beeldscherm:

```
Format complete
A>
Insert diskette for drive A:
and strike a key when ready■
```

Direct na het indrukken van een toets gaat de computer verder met het volgende commando uit het BATCH-programma en laat dat op het beeldscherm zien:

```
A>pause MSX-DOS overzetten?
Strike a key when ready . . . ■
```

Op dit punt kunnen we door <CONTROL>-C in te drukken het BATCH-programma afbreken:

```
A>pause MSX-DOS overzetten?
Strike a key when ready . . . ^C
Terminate batch file (Y/N)? ■
```

Om ons programma af te maken drukken we de letter 'N' in en gaat de computer verder:

```
copy a:command.com b:

Insert diskette for drive B:
and strike a key when ready■
```

Zo zullen alle commando's stap voor stap worden uitgevoerd in de volgorde zoals ze in het BATCH-programma staan, tot en met de weergave van de inhoudsopgave van diskette B. Na het voltooien van het BATCH-programma keert de default cursor weer terug in afwachting van een nieuw commando.
Laten we nog even teruggaan naar de boodschappen op het beeldscherm, bijvoorbeeld:

```
Insert diskette for drive B:
and strike a key when ready■
```

Men zou verwachten dat het indrukken van <CONTROL>-C het BATCH-programma afbreekt, maar in dit stadium zal dit geen invloed hebben en moet men doorgaan met het commando.

```
A>format
Drive name? (A,B) B
Strike a key when ready ■
```

Op deze plaats zal een <CONTROL>-C het commando wel onderbreken dat wordt uitgevoerd en doorgaan met het volgende commando:

```
Drive name? (A,B) B
Strike a key when ready ^C
Aborted
A>copy a:commando.com b:

Insert diskette for drive B:
and strike a key when ready■
```

U ziet, alleen bij PAUSE kunt u het BATCH-programma op een juiste wijze verlaten en stoppen.

Het volgende overzicht bevat een aantal belangrijke aanwijzingen voor het gebruik van BATCH-programma's. Lees deze eerst goed door, voor u zelf een BATCH-programma gaat maken.

* Gebruik nooit de filenaam BATCH (tenzij de naam van het programma dat u wilt uitvoeren BATCH.BAT heet).

* Alleen de filenaam van het uit te voeren BATCH-programma intypen, zonder de extension .BAT.
* Indien u <CONTROL>-C indrukt tijdens het uitvoeren van een BATCH-programma, zal de computer met de volgende vraag komen:

```
Terminate batch file (Y/N)? ■
```

  Toetst u nu 'Y' in, dan zal de computer stoppen met de uitvoering van het BATCH-programma en de default cursor verschijnt weer. Indien u 'N' intoetst, zal de computer verder gaan met het eerstvolgende commando. Het commando dat werd uitgevoerd tijdens het indrukken van <CONTROL>-C, is onderbroken en wordt verder niet uitgevoerd. Gebruik het PAUSE-commando als mogelijkheid om een BATCH-programma te onderbreken en eventueel te verlaten.
* Indien u de diskette met het BATCH-programma tijdens het uitvoeren van dat BATCH-programma uit de diskdrive verwijdert en een andere diskette plaatst, zal de computer vragen om de diskette met het BATCH-programma weer te plaatsen:

```
Insert disk with batch file
and strike any key when ready
■
```

  Ook hier kunt u het BATCH-programma onderbreken met een <CONTROL>-C.
  Zit er helemaal geen diskette in de diskdrive, dan zal de computer de volgende mededeling laten zien:

```
Not ready error reading drive A
Abort, Retry, Ignore? ■
```

  Na het plaatsen van de juiste diskette en het indrukken van de 'R', zal het volgende commando worden gelezen en uitgevoerd.
* Het laatste commando in een BATCH-programma kan de naam zijn van een ander BATCH-programma. Hiermee roept u een volgend BATCH-programma op, zodra het vorige BATCH-programma is beëindigd.

8.2 AUTOEXEC.BAT-programma's

Een bijzondere vorm van een BATCH-programma is het AUTOEXEC.BAT-programma, dat zoals de naam reeds aangeeft, 'automatisch wordt uitgevoerd'. Nadat MSX-DOS is opgestart, kijkt de computer altijd eerst of er op de diskette een program-

ma staat met de naam AUTOEXEC.BAT. Is een dergelijk programma aanwezig, dan zal de datuminvoer worden genegeerd en het AUTOEXEC.BAT-programma ten uitvoer worden gebracht. Deze automatische uitvoering van een BATCH-programma is erg gemakkelijk indien men een diskette heeft met daarop één belangrijk programma dat u automatisch wilt laten RUNnen na het opstarten.
Vindt de computer geen AUTOEXEC.BAT-programma, dan wordt er naar de datum en eventueel de tijd gevraagd.
Een AUTOEXEC.BAT-programma kan natuurlijk ook refereren naar een andere diskette of diskdrive, terwijl het automatisch laten RUNnen van een BASIC-programma na het opstarten ook mogelijk is.

8.3 Het maken van een BATCH-programma

Het maken van een BATCH-programma kan op twee manieren gebeuren. Ten eerste met behulp van het COPY-commando. Heeft u een tekstverwerkend programma, dan is het makkelijker om dat te gebruiken, daar u dan correctiemogelijkheden heeft. Wij zullen ons hier richten op het COPY-commando, omdat dat een standaard- en altijd aanwezig MSX-DOS-commando is.

Allereerst dienen we de computer te vertellen dat we een programma gaan maken met bijvoorbeeld de naam AUTOEXEC.BAT. Tevens moet de computer alle commando's die we invoeren ook in dat programma opslaan en op diskette zetten. Het BATCH-programma kan immers alleen vanaf de diskette worden gelezen en uitgevoerd, de computer leest dan de commando's één voor één in en voert ze ook één voor één uit en wel in de volgorde waarin ze in het programma zijn opgenomen.
Om alles vanaf het toetsenbord naar de diskette te kopiëren gebruiken we dus het COPY-commando:

```
A>COPY CON AUTOEXEC.BAT
■
```

We gaan nu het eerste commando invoeren:

```
A>COPY CON AUTOEXEC.BAT
BASIC■
```

Zoals u weet, zal dit commando ervoor zorgen dat de computer MSX-DOS verlaat en in MSX Disk BASIC komt. Achter het commando BASIC komt nu de naam van het programma dat u automatisch wilt laten RUNnen (vergeet de spatie niet):

```
A>COPY CON AUTOEXEC.BAT
BASIC MENU.BAS■
```

Het programma 'MENU.BAS' dient natuurlijk wel op de diskette te staan waarop ook het BATCH-programma is weggeschreven.
Daar het AUTOEXEC.BAT-programma onder MSX-DOS wordt uitgevoerd, is het ook mogelijk om voor de programmanaam een andere diskette op te geven:

```
A>COPY CON AUTOEXEC.BAT
BASIC B:MENU.BAS■
```

Nu mag het BATCH-programma natuurlijk niet op dezelfde diskette staan.
Hierna zijn er geen commando's meer nodig en sluiten we de commandoregel af met een <RETURN>. Om de commandoregel op diskette te zetten, dienen we <CONTROL>-Z in te drukken, gevolgd door een <RETURN>. Het AUTOEXEC.BAT-programma wordt nu naar diskette weggeschreven.

```
A>COPY CON AUTOEXEC.BAT
BASIC MENU.BAS
^Z
```

Bij het opstarten van MSX-DOS (met de diskette voorzien van dit BATCH-programma), zal het AUTOEXEC.BAT-programma worden uitgevoerd. De computer gaat dan over naar BASIC en zal het programma MENU.BAS automatisch RUNnen.

LET OP:
Gaat men via een AUTOEXEC.BAT-programma over naar BASIC, dan dient het te runnen programma na een spatie direct achter het BASIC-commando te komen. Een ander scheidingsteken kan leiden tot de foutmelding:

```
Bad command or file name
of:
Bad file name
A>■
```

Denk eraan dat de aanwezigheid van een AUTOEXEC.BAT-programma de datuminvoer overslaat. Daar zowel MSX-DOS als Disk BASIC de datum gebruiken bij het wegschrijven van files naar diskette, is het verstandig om in een AUTOEXEC.BAT-programma ook het commando DATE op te nemen. Het AUTOEXEC.BAT-programma ziet er dan bijvoorbeeld zo uit:

```
A>COPY CON AUTOEXEC.BAT        <RETURN>
DATE                           <RETURN>
BASIC MENU.BAS                 <RETURN>
^Z                             <RETURN>
```

Na de laatste <RETURN> zal het programma op diskette worden gezet en geeft de computer de melding:

```
    1 file copied
A>■
```

Door het opnemen van het DATE-commando zal de computer om de datum vragen alvorens het BATCH-programma verder uit te voeren.
Het verdient dan ook aanbeveling om het commando DATE altijd als eerste commando op te nemen in een AUTOEXEC.BAT-programma.

Het eerder genoemde programma NIEUW.BAT komt natuurlijk in aanmerking om op de systeemdiskette te zetten. Hiermee kan men dan snel en eenvoudig nieuwe diskettes formatteren en voorzien van de MSX-DOS-systeemfiles.
Hiertoe typen we eerst het COPY-commando in, gevolgd door



de afzonderlijke opdrachten zoals we ze in paragraaf 8.1 hebben besproken, zie verder Appendix F.

We zien in de meeste handleidingen die bij de computer worden geleverd een programma met de naam 'eeuwigdurende kalender'. Ook in de verschillende boeken met voorbeeldprogramma's komen we dat programma tegen. Onder MSX-DOS is een dergelijk programma zeer eenvoudig te maken, maar het is dan ook geen echte eeuwigdurende kalender. Met het volgende programma zijn alle dagen tussen 1980 en 2099 snel te vinden. De werking berust op het MSX-DOS-commando 'DATE', typ het volgende in:

```
A>COPY CON DATUM.BAT
DATE
DATE
DATUM
^Z
   1 file copied
A>■
```

Met dit eenvoudige BATCH-programma krijgt u tevens een goede indruk van de werking van BATCH-programma's.
Laten we het programma eens uitproberen door het volgende in te typen:

```
A>DATUM
```

Nadat de computer het BATCH-programma heeft ingelezen zien we het volgende op het beeldscherm verschijnen:

```
A>date
Current date is Sun 1-01-84
Enter new date: ■
```

U ziet het, 1 januari 1984 was op een zondag. Voeren we nu een andere datum in, dan zal het BATCH-programma naar de volgende 'commandoregel' gaan. Daar staat weer het commando 'DATE'. Veronderstel dat we als datum hebben ingevoerd 1-1-85, op het beeldscherm verschijnt dan:

```
A>date
Current date is Tue 1-01-1985
Enter new date: ■
```

Dus 1 januari 1985 was op een dinsdag. Om het programma verder te controleren voeren we nog éénmaal een datum in, 1-1-96:

```
A>datum
A>date
Current date is MON 1-01-1996
Enter new date: ■
```

U ziet dat door het woord 'datum' aan het einde van het BATCH-programma op te nemen, het programma automatisch opnieuw wordt gestart na het invoeren van de datum en een <RETURN>. Het programma bevindt zich in een eindeloze lus die alleen kan worden onderbroken met <CONTROL>-C of door het verwijderen van de diskette uit de diskdrive.

8.4 BATCH-programma met variabelen

Het kan voorkomen dat u een serie commando's steeds moet herhalen; steeds met andere filenamen, bijvoorbeeld het toevoegen van een tekstfile aan een reeds bestaande tekstfile.
MSX-DOS heeft voor het gebruik van variabelen in een BATCH-programma 10 verschillende parameters: %0 tot en met %9. Deze kunnen, indien nodig, worden vervangen door filenamen of getallen.
Het BATCH-programma ziet er dan bijvoorbeeld als volgt uit:

```
A>COPY CON TEKST.BAT
COPY %1.TXT LST
COPY ALLE.TXT + %1.TXT
DEL %1.TXT
^Z
   1 file copied
A>■
```

Het BATCH-programma TEKST.BAT bevat maar één variabele, die we straks gaan vervangen door een filenaam. Het uitvoeren van het BATCH-programma gaat met het volgende commando:

```
TEKST INHOUD
```

Het eerste woord is de naam van het BATCH-programma, waardoor deze van diskette wordt gelezen, het tweede woord is de naam die in de plaats van variabele %1 wordt gezet.
Voor de computer ziet het BATCH-programma er als volgt uit:

COPY INHOUD.TXT LST	De tekstfile INHOUD.TXT wordt afgedrukt door de printer.

COPY ALLE.TXT + INHOUD.TXT

De tekstfile INHOUD.TXT wordt gekopieerd en achter de tekstfile ALLE.TXT opgenomen.

DEL %1.TXT

De tekstfile INHOUD.TXT wordt van de diskette verwijderd.

Na het maken van een andere tekstfile, kunnen we met behulp van dit zelfde BATCH-programma die nieuwe tekstfile (bijvoorbeeld 'NWTEKST.TXT') weer toevoegen aan de bestaande tekstfile ALLE.TXT:

```
TEKST NWTEKST
```

Ook nu zal de file eerst door de printer worden afgedrukt en daarna worden opgenomen achter de tekstfile ALLE.TXT. Ten slotte wordt de file gewist.

De variabele %0 is de referentie naar de naam van het BATCH-programma zelf, zonder extension. Als voorbeeld hebben we het volgende BATCH-programma met de naam TEST.BAT:

```
TYPE %2.TXT
COPY %1.TXT LST
TYPE %0.BAT
DIR %3:
```

Alle variabelen dienen in numerieke volgorde te worden opgegeven achter de naam van het BATCH-programma. Het in te voeren commando kan er als volgt uitzien:

```
TEST INHOUD TEKST B
 %0   %1     %2   %3
```

Het ingevoerde commando en de variabelen staan dus in numerieke volgorde, de volgorde die in het BATCH-programma wordt aangehouden is niet van belang. Als eerste opdracht zal het BATCH-programma de tekstfile TEKST.TXT op het beeldscherm weergeven. Daarna wordt de tekstfile INHOUD.TXT naar de printer gestuurd, vervolgens zien we op het beeldscherm de inhoud van het in uitvoering zijnde BATCH-programma TEST.BAT en ten slotte krijgen we de inhoudsopgave van diskette B op het scherm.
Bij het gebruik van meer variabelen is het dus belangrijk om deze in de juiste volgorde op te geven achter de naam van het BATCH-programma. De naam zelf komt overeen met de varia-

bele %0, het eerstvolgende getal of woord dat we intypen is variabele %1, het volgende %2 ... enz. Brengt men in een BATCH-programma alleen de variabele %5 aan, dan kijkt de computer naar de vijfde variabele achter de opgegeven naam van het BATCH-programma. Heeft men alleen de naam van het BATCH-programma en de benaming voor de variabele ingebracht, dus bijvoorbeeld:

```
TEST INHOUD.TXT
```

dan ziet de computer geen vijfde variabele en zal dat leiden tot de melding:

```
File not found
```

Zorgt men ervoor dat de benaming voor de variabele wel op de vijfde plaats komt te staan, dan doet het BATCH-programma zijn werk, bijvoorbeeld:

```
TEST 1 2 3 4 INHOUD.TXT
```

De computer ziet nu de variabelen %0 tot en met %5, ook al worden de variabelen %0 tot en met %4 niet gebruikt, ze moeten worden opgegeven daar het BATCH-programma anders niet de variabele %5 kan vinden! Variabele %5 wordt nu vervangen door INHOUD.TXT en het BATCH-programma functioneert.

We kunnen nu ook de 'eeuwigdurende kalender' uit de vorige paragraaf aanpassen en voorzien van een variabele. Het programma ziet er dan als volgt uit:

```
A>COPY CON DATUM.BAT
DATE %1
DATE
%0
^Z
   1 file copied
A>■
```

Om de juiste dag te vinden, typen we het volgende in:

```
A>DATUM 1-1-85
```

en op het beeldscherm verschijnt:

```
A>DATUM 1-1-85

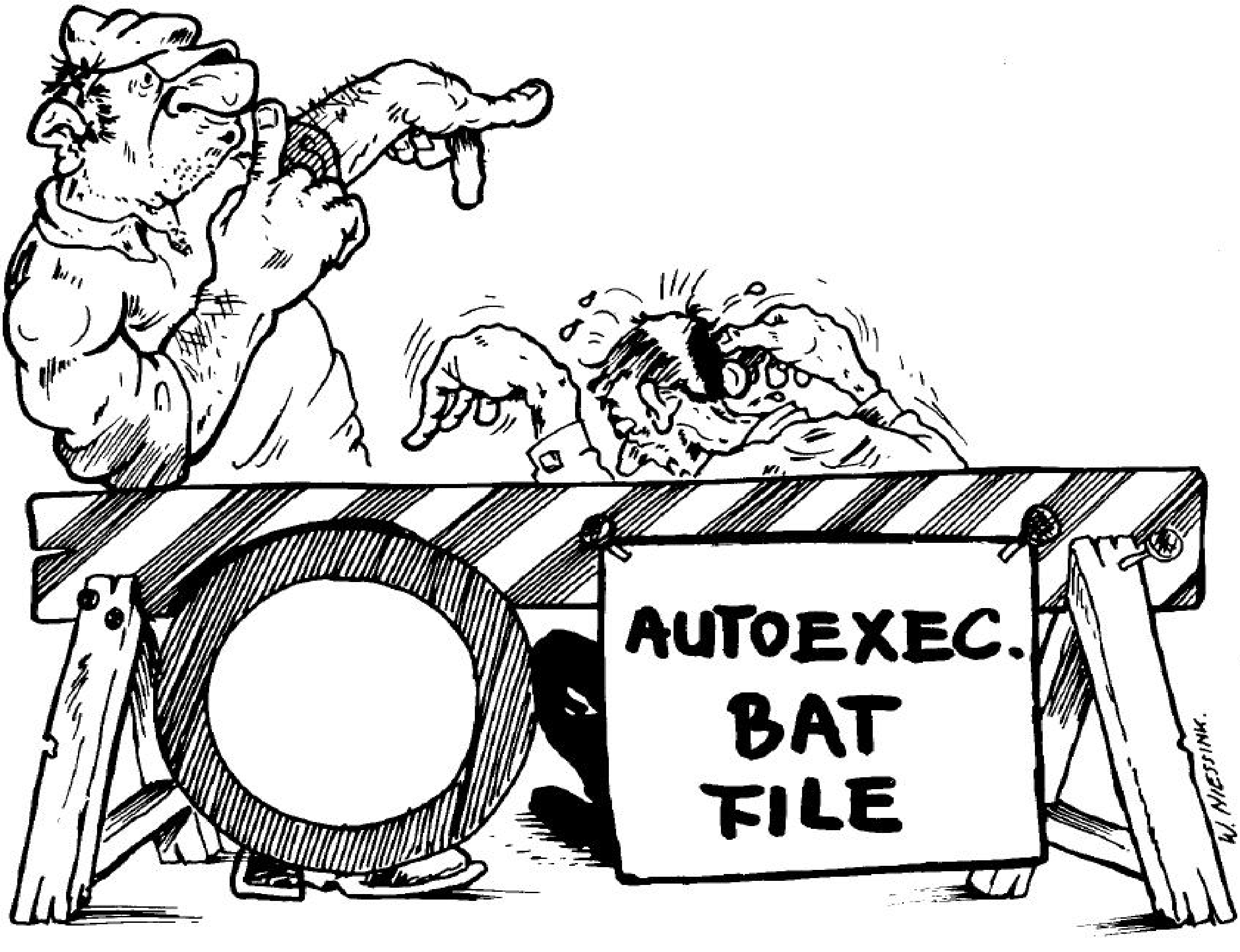
A>date 1-1-85
```

```
A>date
Current date is Tue 1-01-85
Enter new date: ■
```

Na het invoeren van een andere datum gaat het BATCH-programma verder met de volgende opdracht '%0'. Deze variabele komt overeen met de naam van het BATCH-programma en zal dit programma weer opnieuw starten.
Door in plaats van een nieuwe datum '^C' in te voeren, kunnen we het BATCH-programma verlaten.

8.5 AUTOEXEC.BAS-programma's

Ook Disk BASIC kent een programma dat automatisch wordt gerund na het opstarten in Disk BASIC. Denk eraan dat de twee MSX-DOS-systeemfiles zich in dat geval niet op de diskette mogen bevinden.
Een AUTOEXEC.BAS-programma kunt u zowel onder MSX-DOS maken als onder MSX Disk BASIC, maar ook met een tekstverwerkend programma!



Alle regels die gelden voor het schrijven van normale BASIC-programma's gelden ook voor het AUTOEXEC.BAS-programma. U kunt dus een belangrijk programma in BASIC SAVEn onder de naam 'AUTOEXEC.BAS' en dat programma wordt automatisch opgestart na het aanzetten van de computer.
Het is niet mogelijk om vanuit MSX-DOS met een AUTOEXEC.BAT-programma dat alleen het commando 'BASIC' bevat naar een AUTOEXEC.BAS-programma te gaan! U moet dan altijd achter het commando BASIC na de spatie de programmanaam opnemen, het heeft in dit soort gevallen dan ook geen zin om een AUTOEXEC.BAS-programma te schrijven.
Een voorbeeld van een AUTOEXEC.BAS-programma is een programma dat direct na het opstarten de inhoudsopgave laat zien en dan de vraag stelt welk programma u wilt laden. We gaan ervan uit dat het programma onder MSX-DOS wordt gemaakt, hoewel het natuurlijk ook gewoon in Disk BASIC mogelijk is.

```
A>COPY CON AUTOEXEC.BAS
10 files
20 print
30 input''welk programma wenst u'';f$
40 run f$
^Z
   1 file copied
A>■
```

Na het opstarten zonder dat de twee MSX-DOS-systeemfiles op de diskette staan, zult u eerst de inhoudsopgave op het beeldscherm zien met daaronder de vraag 'welk programma wenst u?'. Na het intypen van het gewenste programma (vergeet de eventuele extension niet) en een <RETURN>, wordt dat programma automatisch gerund.

HOOFDSTUK 9

MSX-DOS-EDITING

9.1 Het commandogeheugen

Alle ingetypte commando's worden door het indrukken van de <RETURN>-toets in een commandogeheugen opgeslagen. Dit commandogeheugen heet in het Engels 'template', een term die we hier bewust vermelden, omdat u hem in diverse publicaties kunt tegenkomen.
In eerste instantie zal de computer het commando natuurlijk uitvoeren, maar daar het commando zich ook in het commandogeheugen bevindt, ligt het voor de hand dat het commando met speciale toetsen weer kan worden opgeroepen. Na dat oproepen komt het commando uit het commandogeheugen en wordt op het beeldscherm weergegeven. Drukt men nu de <RETURN>-toets in, dan zal dat commando nogmaals worden uitgevoerd. Blijkt tijdens het uitvoeren van een commando dat er een fout in zit, dan kan men met behulp van speciale correctiefuncties het commando herhalen met een minimum aan typwerk. Als u die functies op de juiste wijze gebruikt, hoeft u alleen de foutieve karakters over te typen.
Na het corrigeren en het laten uitvoeren van een commandoregel, wordt deze commandoregel opgeslagen in het commandogeheugen. Zo zal dus steeds de laatstgegeven opdracht in het commandogeheugen beschikbaar zijn om eventueel in een volgend commando te laten functioneren.

9.2 De functietoetsen

Eerst kijken we naar de cursorbesturingstoetsen, die een deel van de correctiefuncties voor hun rekening nemen. Het beste kan men een commandoregel invoeren en die opslaan in het commandogeheugen door <RETURN> in te drukken. Met deze regel kunnen we oefenen om zodoende de werking van de cursorbesturingstoetsen te leren kennen. De toetsen hebben de volgende functies:

↓ Toont de gehele inhoud van het commandogeheugen.
↑ Verwijdert de commandoregel, die wordt weergegeven, van het beeldscherm.
→ Toont het eerstvolgende karakter uit het commandogeheugen.
← Verwijdert het laatst weergegeven karakter.

Tevens heeft men nog de beschikking over een aantal bijzondere functies om gedeelten van de commandoregel uit het commandogeheugen te halen, om nieuwe commando's toe te voegen en om de nieuwe commandoregel op te slaan in het commandogeheugen zonder deze eerst uit te laten voeren. Met deze laatste functie kan men het commando eerst controleren en eventueel nogmaals corrigeren. Bent u ervan overtuigd dat de regel goed is, dan de ↓-toets indrukken gevolgd door een <RETURN> en het commando wordt ten uitvoer gebracht.
De volgende toetsen geven nog meer mogelijkheden om de inhoud van het commandogeheugen te manipuleren:

SELECT <karakter> Deze functie toont de inhoud van het commandogeheugen tot het opgegeven karakter. Het opgegeven karakter wordt niet getoond.

DEL Wist de karakters in het commandogeheugen. Eén keer indrukken en daarna de ↓-toets, laat de commandoregel zien zonder het eerste karakter. Op identieke wijze worden bij vijf keer indrukken, vijf karakters gewist.

CLR <karakter> Springt over de karakters van de commandoregel in het commandogeheugen tot het opgegeven karakter. Het indrukken van de ↓-toets laat de rest van de commandoregel zien.

INS INSert mode, invoegen. Hierdoor is het mogelijk om karakters of gehele opdrachten tussen karakters te voegen van de commandoregel die zich in het commandogeheugen bevindt.

HOME Door het indrukken van de HOME-toets wordt de gecorrigeerde commandoregel in het commandogeheugen opgeslagen. Op het beeldscherm wordt dat kenbaar gemaakt door achter de nieuwe opgeslagen commandoregel een '@' te zetten, waarna de cursor onder het eerste karakter van die regel komt te staan, zonder de normale 'A>' voor de cursor. U kunt de commandoregel indien nodig weer wijzigen.

De hier beschreven correctiefuncties zijn ook op te roepen met behulp van 'controlekarakters'. Voor sommige functies geldt zelfs meer dan één controlekarakter of meer dan één toets.

Om de verschillende functies te verduidelijken typen we het volgende commando in en geven een <RETURN>:

`DIR A:PROGRAM.COM` <RETURN>

Op het beeldscherm zien we nu alle informatie over de desbetreffende file. Maar het gaat ons om de commandoregel die nu in het commandogeheugen is opgeslagen. Om hetzelfde commando te herhalen, hoeft u slechts de ↓-toets en <RETURN> in te drukken, waarna het op het beeldscherm wordt weergegeven. Na het indrukken van de <RETURN>-toets gaat het commando nogmaals naar de commandoprocessor om te worden uitgevoerd.
Wilt u nu bijvoorbeeld alle gegevens van de file PROGRAM.TXT zien ook op diskette A, dan kunt u gebruik maken van het vorige commando met behulp van de volgende functies, eerst drukken we op:

<SELECT>C `DIR A:PROGRAM.■`

Rechts ziet u het deel uit het commandogeheugen dat op het beeldscherm verschijnt na het indrukken van <SELECT> en de letter C. Nu typt u alleen de letters TXT en de commandoregel ziet er (op het beeldscherm) als volgt uit:

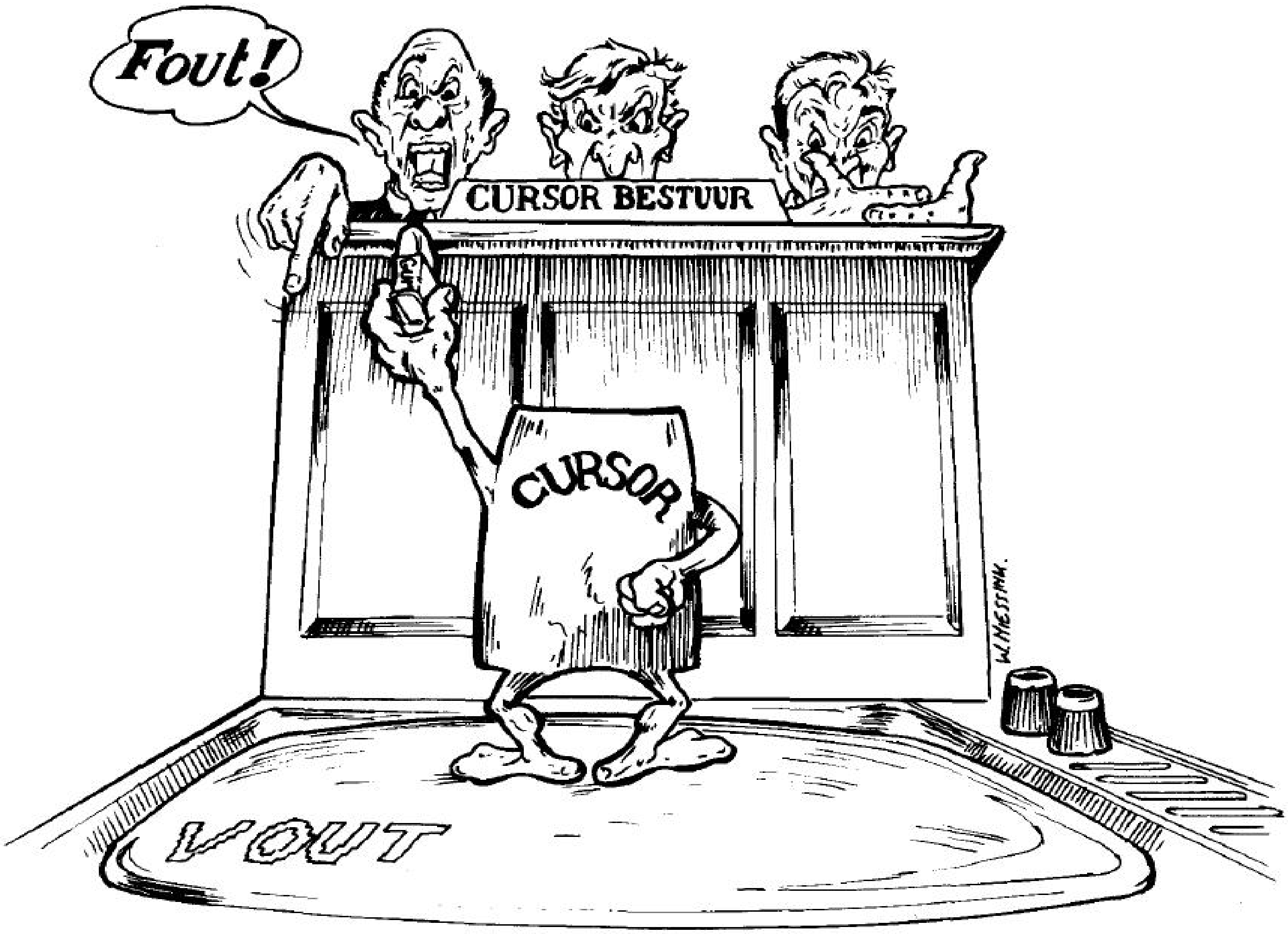
`DIR A:PROGRAM.TXT■`

De commandoregel is nu gereed om te worden doorgestuurd naar de commandoprocessor. Dit doen we met een <RETURN>, waardoor de commandoregel eveneens wordt opgeslagen in het commandogeheugen.

Veronderstel dat we de 'inhoud' van een programma met dezelfde naam dat ook op diskette B staat, nu op het beeldscherm willen zien; we gaan dan als volgt te werk (links de handelingen op het toetsenbord en rechts de weergave op het beeldscherm):

Toetsenbord	*Beeldscherm*
1) TYPE	A>TYPE■
2) <INS>	A>TYPE■
3) <SPATIE>B	A>TYPE B■
4) <DEL>	A>TYPE B■
5) < ↓-toets>	A>TYPE B:PROGRAM.TXT■
6) <RETURN> of <HOME>	

Wat is er nu gebeurd? Allereerst werd in het commandogeheugen het woord 'TYPE' over het woord 'DIR' getypt. Aangezien



er een spatie tussen het eerste commando en de filenaam hoort en deze door de 'E' van **TYPE** is overschreven, drukken we 'INS' in en typen een '<SPATIE>' en vast de letter 'B' voor de andere diskette. In het commandogeheugen staat de cursor nog steeds voor de 'A', die we verwijderen door het indrukken van '<DEL>'. Met de 'cursor omlaag'-toets komt de rest uit het commandogeheugen achter de tekst op het beeldscherm. Het commando ziet er nu uit zoals wij het wensen. Om het nog duidelijker te maken, laten we het gebeuren nogmaals stap voor stap zien - links zoals wij het op het beeldscherm zien - en rechts het gebeuren in het commandogeheugen:

Beeldscherm	*Commandogeheugen*
	DIR A:PROGRAM.TXT
1) TYPE■	TYPE■A:PROGRAM.TXT
2) TYPE■	TYPE■A:PROGRAM.TXT
3) TYPE B■	TYPE B■A:PROGRAM.TXT
4) TYPE B■	TYPE B■:PROGRAM.TXT
5) TYPE B:PROGRAM.TXT■	TYPE B:PROGRAM.TXT■
6) <HOME> of <RETURN>	

Wat u intypt komt over de commandoregel in het commandogeheugen, tot u <INS>ert indrukt. Wat u daarna intypt, wordt gevoegd tussen hetgeen u heeft overschreven in het commandogeheugen en hetgeen er nog staat. Met <DEL>ete wist u het eerstvolgende karakter in het commandogeheugen en met de ↓-toets brengt u de rest uit het commandogeheugen over naar de commandoregel op het beeldscherm.
Door <RETURN> in te drukken, brengt u de commandoregel over naar de commandoprocessor die voor de verdere uitvoering van het commando zorgdraagt. Wilt u het commando niet laten uitvoeren, of denkt u dat er nog een fout in zit, druk dan op de <HOME>-toets om de commandoregel voorlopig in het commandogeheugen op te slaan. U kunt de regel nu rustig nakijken en indien u geen fouten ontdekt, kunt u het commando laten uitvoeren door eerst de ↓-toets in te drukken, waardoor de commandoregel wéér op het beeldscherm verschijnt en daarna de <RETURN>-toets.
Een tweede manier om bovenstaand commando te wijzigen, is de volgende:

Toetsenbord	*Beeldscherm*
1) <INS>	A>■
2) TYPE B	A>TYPE B■
3) <CLR>:	A>TYPE B■
4) <↓-toets>	A>TYPE B:PROGRAM.TXT■
5) <HOME> of <RETURN>	

U ziet dat deze manier minder handelingen vraagt, doch het gaat erom wat voor u in de praktijk het makkelijkste werkt. De enige manier om het manipuleren goed onder de knie te krijgen, is het oefenen met diverse commandoregels.
Laten we aannemen dat we de vorige correctie niet goed hebben uitgevoerd en dat het commandogeheugen de volgende regel laat zien:

```
BYTE B:PROGRAM.TXT
```

Om dit te corrigeren, kunnen we als volgt te werk gaan:

Toetsenbord	*Beeldscherm*
1) <DEL>	A>■
2) <DEL>	A>■
3) <→-toets>	A>T■
4) <INS>	A>T■
5) YP	A>TYP■
6) <↓-toets>	A>TYPE B:PROGRAM.TXT■
7) <HOME> of <RETURN>	

Dit is alleen een voorbeeld om te tonen welke functies de verschillende toetsen uitoefenen. U ziet hier duidelijk dat het indrukken van de <DEL>-toets geen effect heeft op het beeldscherm. Het effect is alleen merkbaar in het commandogeheugen, hetgeen wij niet kunnen waarnemen. Dit geldt natuurlijk ook voor de <INS>-, <SELECT>- en <CLR>-toets. In de praktijk zal men een dergelijke fout op de volgende manier sneller en gemakkelijker kunnen corrigeren:

Toetsenbord	*Beeldscherm*
1) <INS>	`A>■`
2) TYPE	`A>TYPE ■`
3) <CLR>	`A>TYPE ■`
4) B	`A>TYPE ■`
5) < ↓ -toets>	`A>TYPE B:PROGRAM.TXT■`
6) <HOME> of <RETURN>	

U ziet dat er verschillende mogelijkheden zijn om de commandoregel in het commandogeheugen te corrigeren. Het eenvoudigst is het werken met de <CLR>- en de <SELECT>-toetsen, waarbij de éérste over de karakters in het commandogeheugen heen springt en alles achter de denkbeeldige cursor laat staan om met de ↓ -toets naar het beeldscherm over te brengen.
Met <SELECT> brengt u alle karakters over naar het beeldscherm tot het karakter waar u de denkbeeldige cursor in het commandogeheugen heeft geplaatst.

De volgende tabel geeft alle correctiefuncties inclusief de controlekarakters waarmee dezelfde functies kunnen worden uitgevoerd.

Op het beeldscherm zichtbare functies:

↓ ^—	Toont de gehele inhoud van het commandogeheugen.
↑ ESC ^^ ^U ^[	Wist het gehele tot nu toe weergegeven commando van het beeldscherm, laat de inhoud van het commandogeheugen ongewijzigd.
→ ^\	Toont het eerstvolgende karakter uit het commandogeheugen.
← ^H ^]	Wist het laatst weergegeven karakter op het beeldscherm, ingedrukt houden wist de gehele commandoregel karakter voor karakter.
HOME ^K	Brengt de commandoregel vanaf het beeldscherm over naar het commandogeheugen.

Niet zichtbare functies:

SELECT ^X	Brengt alles uit het commandogeheugen over naar het beeldscherm tot aan het gespecificeerde karakter.

CLS ^L	Springt in het commandogeheugen over de karakters tot aan het gespecificeerde karakter.
DEL	Wist het eerstvolgende karakter in het commandogeheugen.
INSERT ^R	Schakelt de invoeg-'mode' in, het commandogeheugen maakt ruimte voor de tussen te voegen tekst.

Noot:
Het '^'-teken geeft aan dat de <CTRL>-toets <CONTROL> ingedrukt dient te worden tegelijk met het daarachter gegeven karakter. Bij '^^' betekent het <CONTROL>-^.

9.3 Functies van de overige controlekarakters

Behalve de hiervoor genoemde controlekarakters is er nog een vijftal dat speciale aandacht verdient. De belangrijkste is <CONTROL>-C, die de commandoprocessor onderbreekt en het in uitvoering zijnde commando stopt. We zijn dit controlekarakter ook tegengekomen bij de BATCH-programma's om na een uitgevoerd commando het BATCH-programma te stoppen, indien we de resterende commando's niet meer willen uitvoeren.
Overal waar u in dit boek de opmerking tegenkomt omtrent het indrukken van een willekeurige toets, is het indrukken van <CONTROL>-C niet inbegrepen.

<CONTROL>-S stopt de weergave op het beeldscherm. Bekijken we bijvoorbeeld met het TYPE-commando een lange tekstfile, dan is de eerste regel van die tekst in een mum van tijd van het beeldscherm verdwenen. Het indrukken van <CONTROL>-S stopt de weergave tot u een willekeurige toets indrukt. Door <CONTROL>-C in te drukken wordt het commando afgebroken en keert de default cursor terug.
Het is handig om <CONTROL>-S als aan/uit-schakelaar te gebruiken bij de weergave van lange stukken tekst e.d. op het beeldscherm. De eerste keer indrukken van <CONTROL>-S zal de weergave stoppen en de tweede keer indrukken van <CONTROL>-S zal de weergave op het beeldscherm continueren.

<CONTROL>-J stuurt de cursor naar het begin van de volgende beeldschermregel. Dit is vooral handig bij lange commandoregels, om meer zicht te krijgen op de afzonderlijke delen van het commando. Met <CONTROL>-J zal alléén het beeldscherm worden beïnvloed, de inhoud van het commandogeheugen verandert daarbij niet. Het gebruik van deze functie kan ertoe leiden dat men scheidingstekens vergeet. Wen er dus aan, om altijd eerst de eventueel vereiste spatie in te voeren voordat u <CONTROL>-J gebruikt.

<CONTROL>-P zet de printer aan (indien aangesloten) en zal dan - als een soort echo - alle informatie die op het beeldscherm komt ook op papier zetten. Deze functie is met name erg handig om de inhoudsopgave van een diskette op papier uit te draaien. Bij het gebruik van deze functie zal in de praktijk blijken dat het geven van een <CONTROL>-P voor het DIR-commando het meest effectief is, daar men dan ook het commando op papier krijgt. Met <CONTROL>-P achter het DIR-commando zal alleen de inhoudsopgave op papier worden gezet en heeft men geen referentie omtrent de diskette waar de inhoudsopgave vanaf is gekomen.

COPY TEKST.TXT LST	is equivalent aan: <CONTROL>-P TYPE TEKST.TXT
<CONTROL>-P DIR B:	zet de inhoudsopgave op papier
COPY DIR B: LST	geeft de foutmelding 'File not found'

<CONTROL>-N schakelt de printer weer uit en het beeldscherm zal de weergavefunctie weer overnemen. Deze functie werkt alleen tussen de commando's door, dus na afloop van een commando kan men de printer uitzetten. Tijdens de uitvoering van een programma of commando is dat niet mogelijk. Vergeet dus niet de printer op tijd uit te zetten.

HOOFDSTUK 10

SAMENVATTING VAN DE COMMANDO'S

10.1 Disk BASIC-commando's en functies

APPEND Aanvullen van een bestand.

```
OPEN "<bestandsnaam>" FOR APPEND AS <bestandsnummer>
```

Een reeds bestaand sequentieel bestand kan worden aangevuld met dit commando. Het aanvullen gaat net als bij het creëren van een bestand.

COPY TO Kopiëren van files.

```
COPY "<drive><filenaam>" TO "<drive><filenaam>"
          1       2                3       4
```

1) Indien er geen diskdrive wordt opgegeven, gaat de computer uit van de default diskdrive of diskette.
2) De filenaam moet volledig, dus inclusief extension worden opgegeven. Het gebruik van wildkarakters is toegestaan, '*.*' zal dus alle files kopiëren.
3) De kopie van de file gaat naar de genoemde diskdrive of diskette. Wordt deze aanduiding weggelaten, dan zal de kopie op de default diskdrive of diskette komen. In het geval dat de te kopiëren file én de kopie beide van en naar één en dezelfde diskette gaan, dient de filenaam afwijkend te zijn. Kopiëren van een file naar dezelfde file kan moeilijkheden geven.
4) Filenaam van de kopie; deze kan worden weggelaten indien de kopie dezelfde naam dient te krijgen als het origineel. Men dient dan wel een andere diskette of diskdrive te specificeren.

COPY TO (2) Kopiëren van files naar andere apparatuur.

```
COPY "<apparaat>" TO "<drive><filenaam>"
          1            2         3
COPY "<drive><filenaam>" TO "<apparaat>"
        4        5              6
```

1) Als apparaat kunnen we alleen CON van 'console' invullen. Na het invoeren van het commando worden alle karakters na het afsluiten met een CTRL-Z naar de gespecificeerde file (3) geschreven. In plaats van een filenaam kunnen we ook hier een apparaat invullen, bijvoorbeeld de printer.
2) Indien er geen diskdrive wordt gespecificeerd zal de kopie naar de default diskdrive of diskette gaan.
3) De nieuwe filenaam moet inclusief de eventuele extension worden opgegeven. Indien men hier de filenaam 'AUTOEXEC.BAS' opgeeft kan men een automatisch startend programma invoeren, zie ook hoofdstuk 8.
4) Bij het niet benoemen van een diskdrive of diskette richt het systeem zich op de default diskette, anders op de diskette die wordt benoemd.
5) Op deze plaats is de volledige filenaam, inclusief extension verplicht. Bij gebruik van wildkarakters geldt hetzelfde als bij het gewoon kopiëren van files.
   `COPY "*.TXT" TO "CON"`
   laat dus alle tekstfiles achter elkaar op het beeldscherm zien.
6) Hier wordt aangegeven naar welk aangesloten apparaat de file dient te worden gekopieerd. Dit kan het beeldscherm (CON) of de printer (LST) zijn. Denk er aan dat alleen ASCII-files kunnen worden weergegeven.

FIELD Voor het dimensioneren van de veldgrootte bij een Random Acces File.

```
FIELD<bestandsnummer>,<veldlengte>AS<veldnaam>,...
            1               2            3
```

1) Het bestandsnummer moet gelijk zijn aan het nummer van het geopende bestand.
2) De lengte per veld wordt opgegeven in bytes. Gebruiken we niet alle bytes, dan worden de 'lege' bytes voorzien van een spatie.
3) De veldnaam moet altijd een stringvariabele zijn, ook al gebruiken we het veld alleen om getallen op te slaan.

Noot:
Na het eerste veld kunnen we een komma opnemen en een volgend veld dimensioneren, tot we alle velden hebben benoemd.

FILES Geeft de inhoudsopgave van de diskette.

```
FILES "<drive><filenaam>"
          1        2
```

1) Indien u de inhoudsopgave van een andere dan de default diskette wilt zien dient u hier de gewenste diskdrive of diskette op te geven.
2) De filenaam is een optie, die u alleen nodig heeft bij het gebruik van wildkarakters.

Noot:
Bij het gebruik van het commando LFILES wordt de inhoudsopgave naar de printer gestuurd. Verder is het gebruik gelijk aan FILES.

FORMAT Hiermee maakt u de diskettes gereed voor MSX Disk BASIC. Na het commando krijgt u de gelegenheid om een andere diskette te plaatsen en desgewenst een diskdrive op te geven.

```
CALL FORMAT of _FORMAT
```

GET Brengt de gegevens vanaf de diskette naar de filebuffer.

```
GET <bestandsnummer>,<recordnummer>
          1                2
```

1) Het bestandsnummer dient overeen te komen met het reeds geopende bestand. Lezen in een niet-geopend bestand geeft een foutmelding.
2) Het recordnummer mag niet 'nul' zijn of groter dan het aantal aanwezige records.

KILL Wist de gespecificeerde file van een diskette.

```
KILL "<drive><filenaam>"
         1        2
```

1) Indien de drive niet wordt benoemd, zal de opgegeven file op de diskette in de default diskdrive worden gewist.

2) De filenaam moet inclusief de extension worden opgegeven. Het gebruik van wildkarakters is toegestaan, doch raden we ten zeerste af.

```
KILL "*.*"
```

zal alle files op de default diskette onherroepelijk wissen.

LEN Dimensioneert de lengte van een record bij een Random Acces File, een geheel getal, maar mag niet groter zijn dan 256.

LSET Hiermee brengen we een stringvariabele over van het werkgeheugen naar de filebuffer en wordt de string tevens omgezet naar de juiste variabele die overeenkomt met de variabele die in het FIELD-commando is gedimensioneerd.

NAME Verandert de naam van een gespecificeerde file.

```
NAME "<drive><oude filenaam>" AS "<nieuwe filenaam>"
        1        2                     3
```

1) Het opgeven van een diskdrive is alleen nodig als men een filenaam gaat veranderen van een andere dan de default diskette.
2) De oude filenaam dient men volledig op te geven, inclusief extension. Het gebruik van wildkarakters is toegestaan.
3) De nieuwe filenaam dient volledig te worden opgegeven, inclusief extension. Bij gebruik van wildkarakters in de oude filenaam, dienen deze ook in de nieuwe filenaam te worden gebruikt. Het benoemen van een diskdrive of diskette bij de nieuwe filenaam wordt genegeerd.

PUT Brengt de gegevens vanuit het werkgeheugen over naar de filebuffer.

```
PUT <bestandsnummer>,<recordnummer>
          1               2
```

1) Het bestandsnummer moet overeenkomen met het reeds geopende bestand, anders krijgt men een foutmelding.
2) Het recordnummer kan een willekeurig geheel getal zijn. Het grootst mogelijke recordnummer is '4.294.967.295'.

RSET Hiermee brengen we een stringvariabele over van het werkgeheugen naar de filebuffer en wordt de string tevens omgezet

naar de juiste variabele die overeenkomt met de variabele die in het FIELD-commando is gedimensioneerd.

SYSTEM Terugkeren naar MSX-DOS.

```
CALL SYSTEM of
_SYSTEM
```

Met dit commando keert men terug naar MSX-DOS. Het commando is alleen beschikbaar als men heeft opgestart met MSX-DOS en met het commando BASIC naar Disk BASIC is gegaan.

VERIFY Voor het controleren van de weggeschreven gegevens.

```
CALL VERIFY ON of
_VERIFY ON
```

Alle gegevens die naar de diskette worden geschreven, worden tevens gecontroleerd aan de hand van de gegevens in het werkgeheugen. Ontdekt de computer verschil, dan verschijnt er een foutmelding en dient men de gegevens opnieuw naar de diskette te schrijven.

CVI – CVS – CVD Deze functies zetten een stringvariabele uit een Random Acces File om in een numerieke waarde als deze is gemaakt met respectievelijk de commando's MKI$, MKS$ of MKD$.

CVI is voor integers opgeslagen in twee bytes.
CVS is voor enkele precisie opgeslagen in vier bytes.
CVD is voor dubbele precisie opgeslagen in acht bytes.

DSKF Deze functie geeft de vrije ruimte op de diskette aan.

DSKF (diskdrivenummer)

Het diskdrivenummer voor de default diskdrive is 0.

LOC Geeft het laatst gelezen recordnummer.

LOC (bestandsnummer)

Het bestandsnummer dient overeen te komen met een reeds geopend bestand, anders verschijnt er een foutmelding. Gebruikt men deze functie bij een sequentieel bestand, dan krijgt men het totaal aantal ingelezen items.

LOF Geeft de totale lengte van een bestand in bytes.

LOF (bestandsnummer)

Het bestandsnummer moet overeenkomen met een reeds geopend bestand, anders verschijnt er een foutmelding. Bij een sequentieel bestand krijgt men per item 2 bytes meer.

MKI$ – MKS$ – MKD$ Deze functies zetten een numerieke waarde om in een string, waarna de string is op te nemen in een record van een Random Acces File.

MKI$ zet een integer om naar een 2-bytes string.
MKS$ zet een getal met enkele precisie om naar een string van vier bytes.
MKD$ zet een getal met dubbele precisie om naar een string van acht bytes.

10.2 MSX-DOS-commando's

BASIC

1) BASIC
2) BASIC <filenaam>

1) Verlaten van MSX-DOS en overgaan naar MSX Disk BASIC. Om weer terug te keren naar MSX-DOS het commando CALL SYSTEM gebruiken. Denk er wel aan dat in dit laatste geval de default diskette moet zijn voorzien van de twee DOS-files om MSX-DOS weer te kunnen laden. De systeem-'call' maakt deel uit van MSX-DOS, hetgeen betekent dat deze 'call' alleen werkt indien de computer is opgestart met MSX-DOS.
2) Verlaten van MSX-DOS en overgaan naar MSX Disk BASIC en automatisch laten runnen van het gespecificeerde BASIC-programma.

COPY Het COPY-commando bestaat uit de volgende algemene opdracht:

```
COPY <drive><filenaam> <samenvoegen>
        1       2            3
     <drive><filenaam>
        4       5
```

1) Indien er geen drive of diskette wordt opgegeven, zal de file vanaf de default diskette worden gekopieerd.
2) De filenaam moet volledig, dus inclusief extension, worden opgegeven. Het gebruik van wildkarakters is toegestaan. '*.*' zal dus alle files kopiëren.

3) Het koppelen van verschillende ASCII-files kan met behulp van het '+'-teken; in andere gevallen komt er tussen de namen een spatie.
4) De kopie van de file gaat naar de genoemde drive en diskette. Wordt deze aanduiding weggelaten, dan zal de kopie op de default diskette komen. In het geval dat de te kopiëren file én de kopie beide van en naar één en dezelfde diskette gaan, dient de filenaam afwijkend te zijn.
5) Filenaam van de kopie; deze kan worden weggelaten indien de kopie dezelfde naam dient te hebben als het origineel, maar deze dient dan wel op een andere diskette te komen.

COPY (2)

```
COPY <con> <drive><filenaam>
       1      2        3
COPY <drive><filenaam> <con>
        4        5       6
```

1) CON refereert aan het toetsenbord en AUX aan een aangesloten invoerapparaat als invoer.
2) Bij het opgeven van een drive- of diskettebenaming zal alles naar de genoemde diskette worden gekopieerd. Het niet benoemen zal alles naar de default diskette kopiëren.
3) Men dient altijd een filenaam op te geven, anders kopieert het systeem naar zichzelf en staat de tekst nogmaals op het beeldscherm.
4) Bij het niet benoemen van een diskette richt het systeem zich op de default diskette, anders op de diskette die wordt benoemd.
5) Op deze plaats is de volledige filenaam, inclusief extension, verplicht. Bij het gebruik van wildkarakters geldt hetzelfde als bij het gewone kopiëren van files.

```
COPY *.TXT CON
```

laat dus alle tekstfiles achter elkaar op het beeldscherm zien.

6) Hier wordt aangegeven naar welk aangesloten apparaat de file dient te worden gekopieerd. Dit kan een willekeurig (AUX) apparaat zijn of de printer (LST of PRN) en natuurlijk het beeldscherm (CON).

DATE

```
1) DATE
2) DATE <M>-<D>-<Y>
```

1) Voor het controleren en veranderen van de datum.
2) Voor het 'direct' veranderen van de datum.

DEL `DEL` <drive><filenaam>

Indien de drive niet wordt benoemd, zal de opgegeven file op de default diskdrive worden gewist. De filenaam dient altijd inclusief de extension te worden opgegeven, waarbij het gebruik van wildkarakters is toegestaan.

DIR `DIR` <drive><filenaam><mode>
1 2 3

1) Indien u de gegevens van één of meer files wilt zien van een andere dan de default diskette, dient u de gewenste diskette op te geven.
2) De filenaam is een optie, die alleen nodig is als u van een bepaalde file alle gegevens wilt zien. Bij het weglaten van de filenaam zal de gehele inhoudsopgave van de default of opgegeven diskette worden weergegeven. Het gebruik van wildkarakters in de filenaam is toegestaan.
3) U kunt met de toevoeging '/W' de filenamen naast elkaar op het scherm laten weergeven. Met de toevoeging '/P' zullen er niet meer dan 23 files tegelijk worden weergegeven.

FORMAT Hiermee maakt u de diskettes gereed voor MSX-DOS. Na het commando krijgt u de mogelijkheid om een diskette te plaatsen en desgewenst een diskdrive op te geven.

MODE `MODE` <breedte>

Achter dit commando kunt u de gewenste karakterbreedte van het beeldscherm opgeven. Bij een breedte van 32 of minder zal het grafische scherm in werking zijn. Standaardinstelling is de tekstmode met een breedte van 37 karakters.

REN `REN` <drive><oude filenaam> <nieuwe filenaam>
1 2 3

1) De drive die de oude file bevat, dient men op te geven, tenzij de file op de default diskette staat.
2) De file die van naam moet veranderen, dient men volledig op te geven, inclusief extension. Het gebruik van wildkarakters is toegestaan, alle files die aan die voorwaarde voldoen, worden dan van naam veranderd.

3) De nieuwe filenaam dient volledig te worden opgegeven, indien nodig inclusief de extension. Het gebruik van wildkarakters is toegestaan. Op de plaats van de wildkarakters zal de oude filenaam niet veranderen.

In plaats van het commando REN is de volledige benaming RENAME ook toegestaan.

TIME

```
TIME <hh><:mm><:ss><.cc>
       1    2    3    4
```

1) De uren worden opgegeven in één of twee cijfers. Alléén uren invoeren is niet mogelijk en leidt tot een foutmelding, dus:
2) dient men de minuten gescheiden door een dubbele punt in te voeren. Ook voor de minuten één of twee cijfers.
3) De seconden mogen worden weggelaten bij het invoeren van een nieuwe tijd, dan wordt namelijk aangenomen dat 'ss = 00'. Voert men de seconden wel in, dan heeft men een extra optie om honderdsten van een seconde in te voeren, van de hele seconden gescheiden door een enkele punt!

TYPE

```
TYPE <drive><filenaam>
       1        2
```

1) Indien men geen diskdrive opgeeft, zal de file vanaf de default diskette worden gelezen en weergegeven op het scherm.
2) De filenaam dient altijd te worden opgegeven, inclusief de extension. Het gebruik van wildkarakters wordt afgeraden. Het commando neemt dan de eerste file die voldoet aan de specificatie en zal na het uitvoeren van die opdracht stoppen en geen volgende file meer weergeven, ook al voldoet die file aan de opgegeven specificatie met het wildkarakter.

VERIFY

```
VERIFY <on // off>
```

U heeft bij het VERIFY-commando de keuze tussen ON (wel controleren) en OFF (niet controleren). Standaard-instelling is OFF.

10.3 Batch-commando's

PAUSE

```
PAUSE <mededeling>
```

Na het PAUSE-commando in een BATCH-programma wacht het besturingssysteem totdat u een toets indrukt. Via <CONTROL>-C kunt u de uitvoering van het BATCH-programma onderbreken.
Na het weergeven van een eventuele mededeling, vraagt de computer altijd om een toets in te drukken.

REM `REM` <mededeling>

Achter het REM-commando kunt u boodschappen opnemen die in tegenstelling tot BASIC wél op het beeldscherm komen tijdens de uitvoering van het programma.

10.4 Controlekarakters bij MSX-DOS

Karakter Beschrijving functie
^C Onderbreekt commando's en BATCH-programma's.
^H Wist het laatste karakter van een commandoregel op het beeldscherm.
^J Plaatst de cursor aan het begin van de volgende beeldschermregel.
^K Brengt de commandoregel vanaf het beeldscherm over naar het commandogeheugen.
^L Springt in het commandogeheugen over de karakters tot aan het gespecificeerde karakter.
^N Zet de printer uit.
^P Zet de printer aan.
^R Het commandogeheugen maakt ruimte voor het tussenvoegen van karakters.
^S Onderbreekt tijdelijk de weergave op het beeldscherm.
^U Wist de commandoregel van het scherm.
^X Brengt de inhoud van het commandogeheugen over naar het beeldscherm tot het gespecificeerde karakter.
^\ Toont het eerstvolgende karakter uit het commandogeheugen.
^^ Wist de commandoregel van het scherm.
^[Wist de commandoregel van het scherm.
^] Wist het laatste karakter van een commandoregel op het beeldscherm.

HOOFDSTUK 11

APPENDICES

Appendix A Foutmeldingen bij MSX-DOS

Buiten het weergeven van verschillende mededelingen op het beeldscherm, zal MSX-DOS ook fouten tijdens het uitvoeren registreren en daar melding van maken. Voordat MSX-DOS die melding op het beeldscherm weergeeft, probeert het besturingssysteem het commando alsnog uit te voeren, en wel drie keer. Mocht de fout nog steeds blijven bestaan, dan zal MSX-DOS met de volgende melding komen:

```
<1> error <2> drive <3>
Abort, Retry, Ignore? ■
```

1 Op deze plaats staat een mededeling die de mogelijke fout van de diskette of diskdrive aangeeft. De volgende meldingen kunt u verwachten:

`Write protect`	De diskette kan niet worden beschreven.
`Not ready`	Als er geen diskette in de drive is geplaatst.
`Disk`	Treedt o.a. op als de diskette niet is te lezen of te beschrijven, bijvoorbeeld indien de diskette nog niet is geformatteerd.

2 Deze melding zegt iets over uitvoering van het commando op het moment dat de fout optrad. Er zijn twee mogelijkheden:

`reading`	Tijdens het lezen van de diskette.
`writing`	Tijdens het schrijven naar diskette.

3 Ten slotte krijgt u de melding bij welke diskette of diskdrive de fout is opgetreden.

`A, B, C of D` Eén van de diskdrives.

Onder de foutmelding vraagt MSX-DOS wat er met die fout moet gebeuren, het intoetsen van de eerste letter voert het gevraagde commando uit:

A Abort (stoppen). Het programma wordt onderbroken en de default cursor komt terug op het beeldscherm.
R Retry (opnieuw proberen). MSX-DOS zal, nadat u de fout zo mogelijk heeft hersteld, proberen het commando nogmaals uit te voeren.
I Ignore (negeren). MSX-DOS zal een eventuele slechte sector van de diskette waar de fout is ontstaan negeren en doen alsof de fout niet is opgetreden. Hiermee kan wel informatie verloren gaan.

Over het algemeen zal men eerst de 'R' opgeven om te kijken of het een toevallige éénmalige fout was. Treedt de fout weer op, dan toetsen we de 'A' in om te stoppen.
De volgende foutmelding treedt op indien er geen diskette is geplaatst en de computer een file vanaf de diskette wil lezen:

```
Not ready error reading drive A
Abort, Retry, Ignore? ■
```

Welke letter u ook intypt, de computer komt steeds met deze foutmelding terug tot u een diskette heeft geplaatst. Ook na het plaatsen van de diskette maakt het niet uit welke letter u intypt, de computer maakt eerst het commando af waar hij mee bezig was.

Een foutmelding die ook kan ontstaan tijdens het lezen van of het schrijven naar een diskette is:

```
Bad FAT
```

Deze melding geeft aan dat de gegevens in de 'File Allocation Table' verwijzen naar een niet-bestaande sector. Het kan zijn dat de diskette verkeerd of in het geheel niet is geformatteerd. Treedt deze fout op, dan is de informatie op de diskette niet meer te achterhalen en dus onbruikbaar. De enige manier om de diskette weer te kunnen gebruiken is deze opnieuw te formatteren, waarbij alle eventueel aanwezige informatie op de diskette verloren gaat.

Behalve deze twee foutmeldingen kent MSX-DOS nog een aantal foutmeldingen en een aantal mededelingen die regelmatig kunnen verschijnen.

Aborted
Deze uitdrukking verschijnt na het indrukken van ^C op momenten dat de computer vraagt naar het indrukken van een willekeurige toets tijdens een BATCH-programma. Behalve op die plaatsen waar men het BATCH-programma mag en kan onderbreken.

Are you sure (Y/N)? ■
Indien men alle files van de diskette wil wissen, vraagt de computer u eerst om een bevestiging van het ingevoerde commando.

Bad command or file name
Het opgegeven commando bestaat niet, of het opgegeven woord refereert niet aan een bestaande file.

Boot error
Press any key for retry
Deze fout treedt op als men teruggaat naar MSX-DOS zonder de diskette die is voorzien van de twee systeemfiles.

Disk I/O error
Deze foutmelding treedt alleen op als er tijdens het VERIFY ON-commando een fout wordt geconstateerd.

Drive name? (A,B)
Na het FORMAT-commando vraagt de computer welke diskette moet worden geformatteerd.

.. file copied
Zodra er één of meer files zijn gekopieerd, zal de computer daar melding van maken voorafgegaan door het aantal gekopieerde files.

File not found
Deze foutmelding zal optreden als we de inhoudsopgave van een lege diskette opvragen of een niet op diskette staande filenaam opgeven achter een commando.

Format complete
Zodra het formatteren is voltooid, komt deze mededeling op het beeldscherm.

Insert diskette for drive A:
and strike a key when ready ■
De computer geeft steeds aan wanneer er van diskette moet worden gewisseld. Na het indrukken van een willekeurige toets zal het commando worden afgemaakt. Het gebruik van ^C heeft hier geen invloed.

Insert disk with batch file
and strike any key when ready ■
Indien u de diskette met het in uitvoering zijnde BATCH-programma verwijdert, zal deze foutmelding optreden. Het intoetsen van ^C zal het BATCH-programma op dit punt onderbreken.

Invalid date
Enter new date:
Deze mededeling verschijnt indien men tijdens het opstarten of na het commando DATE een verkeerde datum opgeeft.

Invalid drive specification
Het opgeven van de benaming van een niet-aangesloten diskdrive resulteert in deze foutmelding.

Invalid parameter
Deze foutmelding treedt op indien men een verkeerd getal opgeeft, bijvoorbeeld het commando MODE 42.

Invalid time
Enter new time:
Deze mededeling verschijnt indien men tijdens het opgeven van de tijd een fout maakt. Werkt niet bij alle MSX-computers.

Insufficient disk space
Indien de diskette onvoldoende ruimte heeft om de file weg te schrijven, de diskette is dan vol.

Rename error
Indien men een niet op diskette staande file van naam wil veranderen, treedt deze foutmelding op. Ook als de nieuwe naam reeds als filenaam op de diskette voorkomt, treedt deze melding op.

Strike a key when ready . . . ■
Na een PAUSE-commando in een BATCH-programma komt deze mededeling. U kunt dan met ^C het BATCH-programma onderbreken.

Terminate batch file (Y/N)?
Deze boodschap verschijnt indien een BATCH-programma wordt onderbroken met <CONTROL>-C. Een 'Y' zal de BATCH-file doen stoppen, terwijl bij 'N' de BATCH-file begint met het uitvoeren van het eerstvolgende commando.

Appendix B Foutmeldingen en codes bij Disk BASIC

Bad allocation table **err=60**
Als de diskette nog niet is geformatteerd en men de inhoudsopgave opvraagt.

Bad drive name **err=62**
Een niet-aangesloten diskdrive wordt gespecificeerd in één van de commando's.

Bad file mode **err=61**
Bij het gebruik van de commando's PUT, GET of LOF bij een sequentiële file. Openen van een file zonder 'FOR OUTPUT', 'FOR INPUT' of 'FOR APPEND'.

Bad file name **err=56**
Als men een filenaam opgeeft direct achter het MSX-DOS-commando BASIC (dus zonder spatie), verschijnt in Disk BASIC deze foutmelding.

Bad file number **err-52**
Een commando verwijst naar een niet-geopende file of valt buiten het aantal geïnitialiseerde files.

Bad sector number **err=63**
Indien men een verkeerd sectornummer opgeeft bij het rechtstreeks lezen van of schrijven naar de diskette.

Direct statement in file **err=57**
Bij het laden van een ASCII-file die voorzien is van een direct commando. Het laden wordt afgebroken.

Disk full **err=66**
Er is geen ruimte genoeg op de diskette om een programma of een file weg te schrijven.

Disk I/O error **err=69**
Een fout bij het lezen of schrijven van/naar de diskette. De computer kan hierop blijven hangen. Deze foutmelding treedt ook op bij niet-geformatteerde diskettes.

Disk offline err=70
Er bevindt zich geen diskette in de gespecificeerde diskdrive.

Disk write protected err=68
Indien men wil wegschrijven naar een diskette die beveiligd is tegen schrijven.

Field overflow err=50
Deze melding treedt op als men meer bytes wil gebruiken in een 'FIELD' dan in de lengte van een RECORD zijn gespecificeerd.

File already exist err=65
De file die wordt gespecificeerd bestaat reeds op de diskette.

File already open err=54
Indien een file al eerder werd geopend, of als men een open file probeert te wissen met het KILL-commando.

File not found err=53
De opgegeven file staat niet op de default diskette.

File not OPEN err=59
Bij het lezen van of het schrijven naar een niet-geopende file.

File still open err=64
De file is nog niet afgesloten met 'CLOSE'.

Input past end err=55
Er wordt om een INPUT gevraagd terwijl alle data al is ingelezen. Het gebruik van het commando EOF (einde file) voorkomt deze foutmelding.

Internal error err=51
Deze melding zal zelden op uw beeldscherm verschijnen, daar ze alleen ontstaat als er een defect aan uw computer is.

Rename across disk err=71
Bij het veranderen van een filenaam van de ene diskdrive naar de andere.

Sequential I/O only err=58
Het GET- of PUT-commando wordt gebruikt bij een sequentiële file.

Too many files err=67
Als er 255 files (het maximum) op diskette staan en men probeert een programma te SAVEn of een file te openen.

Appendix C Gereserveerde woorden en namen

MSX-DOS kent een aantal gereserveerde woorden. In principe zijn namen van .COM-files en .BAT-programma's ook gereserveerde woorden, die men beter niet kan gebruiken voor data- of programmafiles op dezelfde diskette. Geen enkel MSX-DOS-commando mag men gebruiken als filenaam. Dit zijn de volgende woorden:

AUX	DEL	MODE	REN
BASIC	DIR	NUL	RENAME
CON	ERASE	PAUSE	TIME
COPY	FORMAT	PRN	TYPE
DATE	LST	REM	VERIFY

Buiten deze woorden zijn er nog een aantal volledige filenamen die niet mogen worden gebruikt, omdat deze files een speciale betekenis hebben:

AUTOEXEC.BAS
AUTOEXEC.BAT
COMMAND.COM
MSXDOS.SYS

Tevens dient men te voorkomen dat zelfgemaakte datafiles, bijvoorbeeld adresbestanden, worden voorzien van de extension .COM of .BAT. Deze extensions hebben voor MSX-DOS een speciale betekenis.

Naast de hiervoor genoemde woorden is het verstandig om de gereserveerde woorden die in MSX BASIC gelden niet te gebruiken bij MSX-DOS. Dit om eventuele verwarring te vermijden.

Gereserveerde woorden in Disk BASIC

ABS	DEFDBL	INPUT	NEXTOFF	SCREEN
AND	DEFINT	INPUT$	NOT	SGN
AS	DEFSNG	INSTR	OCT$	SIN
ASC	DEFSTR	INT	OFF	SOUND
ATN	DEFUSR	INTERVAL	ON	SPACE$
AUTO	DELETE	KEY	OPEN	SPC
BASE	DIM	KILL	OR	SPRITE
BEEP	DRAW	LEFT$	OUT	SPRITE$
BIN$	DSKF	LEN	PAD	SQR
BLOAD	ELSE	LET	PAINT	STEP
BSAVE	END	LINE	PDL	STICK
CALL	EOF	LIST	PEEK	STOP
CDBL	EQV	LLIST	PLAY	STR$
CHR$	ERL	LOAD	POINT	STRIG
CINT	ERR	LOC	POKE	STRING$
CIRCLE	ERROR	LOCATE	POS	SWAP
CLEAR	EXP	LOF	PRINT	TAB
CLOAD	FIELD	LOG	PSET	TAN
CLOSE	FILES	LPOS	PRESET	THEN
CLS	FIX	LPRINT	PUT	TROFF
COLOR	FN	LSET	READ	TRON
CONT	FOR	MAXFILES	REM	USING
COS	FRE	MERGE	RENUM	USR
CSAVE	GET	MID$	RESTORE	VAL
CSNG	GOSUB	MKD$	RESUME	VARPTR
CSRLIN	GOTO	MKI$	RETURN	VDP
CVD	HEX$	MKS$	RIGHT$	VPEEK
CVI	IF	MOD	RND	VPOKE
CVS	IMP	MOTOR	Rset	WAIT
DATA	INKEY$	NAME	RUN	WIDTH
DEF	INP	NEW	SAVE	XOR

Appendix D Overeenkomst van en verschil tussen Disk BASIC en MSX-DOS

Overeenkomst van de commando's

Gelukkig bestaan er enkele commando's die zowel in Disk BASIC als in MSX-DOS hetzelfde zijn en die dan ook de minste problemen zullen geven. De belangrijkste zaken die in ieder geval gelijk moeten blijven, zijn de benamingen van de programma- en datafiles. Het moet natuurlijk mogelijk zijn om vanuit MSX-DOS de files op te roepen die in Disk BASIC zijn geschreven. De opbouw van de filenaam is dan ook identiek, zoals het aantal karakters, de extension, de gebruikte karakters, de aanbevolen extensions enz.

Bij het zoeken naar de verschillende files zijn ook dezelfde wild-

karakters toegestaan. De gereserveerde filenamen in Disk BASIC gelden ook voor MSX-DOS.

Eigenlijk zijn er maar twee Disk BASIC commando's die we ook kennen bij MSX-DOS, namelijk FORMAT en COPY. In Disk BASIC dient men het commando FORMAT op te roepen met CALL of verkort met '—'. In MSX-DOS is dat oproepen niet nodig en is alleen het commando al voldoende om een nieuwe diskette te formatteren.
Het COPY-commando (ondanks de overeenkomst) zal zeker in het begin problemen kunnen geven. In Disk BASIC dient men echt te kopiëren van de ene 'naar' (Engels = to) de andere diskette. In MSX-DOS is dat niet nodig daar het systeem weet dat achter het COPY-commando twee namen komen, waarbij de eerste filenaam de naam van de originele file is en de tweede filenaam de kopie moet worden. Tevens worden de bij Disk BASIC noodzakelijke aanhalingstekens onder MSX-DOS geweigerd en leidt dat tot de melding:

```
File not found
```

Bij Disk BASIC mag men achter het COPY-commando, de eerste filenaam, 'TO' en de tweede filenaam achter elkaar opgeven, zonder scheidingsteken. In principe worden de aanhalingstekens al gezien als scheiding tussen de verschillende namen en opdrachten. Bij MSX-DOS dient men beslist gebruik te maken van de <SPATIE> als scheidingsteken (zie paragraaf 7.2 voor andere scheidingstekens). Zonder deze speciale scheidingstekens krijgt men onherroepelijk een foutmelding. De aanhalingstekens kunnen bij MSX-DOS niet worden gebruikt.

Verschil in commando's

Waarschijnlijk veroorzaakt het omschakelen tussen de commando's in MSX-DOS en Disk BASIC de meeste problemen. Eén van de eerste dingen die men wil zien is de inhoudsopgave, die men bij Disk BASIC oproept met het commando 'FILES'. Bij MSX-DOS is dat de afkorting 'DIR' van DIRectory geworden. Alle bestaande varianten van het commando 'FILES', die men kan toepassen bij Disk BASIC zijn ook van toepassing bij MSX-DOS, alleen zonder aanhalingstekens en denk ook aan de <SPATIE> als scheidingsteken tussen het commando en de benaming van de diskette, diskdrive of filenaam.
Bij Disk BASIC is het wissen van files op de diskette mogelijk met het commando 'KILL'. MSX-DOS kent hiervoor twee commando's, namelijk 'DEL' en 'ERASE'. Verder gelden ook hier weer de al eerder genoemde regels betreffende de aanhalingstekens en scheidingstekens.

Het veranderen van een filenaam gebeurt in Disk BASIC met het commando NAME "<filenaam>" AS "<filenaam>". Bij MSX-DOS heet dit commando 'REN' of voluit 'RENAME' en wordt weer opgegeven met spaties als scheidingstekens.
In Disk BASIC kennen we de commando's LOAD en SAVE, die bij MSX-DOS niet meer voorkomen. Het SAVEn gaat met behulp van het COPY-commando en het LOADen door middel van een COMMAND.COM- of AUTOEXEC.BAT-programma. In Disk BASIC kennen we de file AUTOEXEC.BAS. Deze laatste komt dan overeen met het BASIC-commando 'LOAD,R'. Ten slotte kent MSX Disk BASIC het commando 'MERGE', voor BASIC-programma's. Bij MSX-DOS hebben we niet rechtstreeks te maken met BASIC-programma's, maar met commandofiles, die ook zijn te koppelen met een bepaald commando. Het koppelen van commando- en andere files, bijvoorbeeld tekstfiles, gebeurt bij MSX-DOS met het COPY-commando en het '+'-teken.

Een commando dat speciale aandacht verdient en alleen werkt bij MSX Disk BASIC indien u met MSX-DOS heeft opgestart, is het commando CALL SYSTEM. Dit commando maakt het mogelijk om terug te keren naar MSX-DOS als u dat via het commando BASIC heeft verlaten.
Heeft u niet opgestart met MSX-DOS, dan zal de computer het commando niet herkennen. Dat betekent dat MSX-DOS tijdens het opstarten een commando aan 'de woordenlijst' van MSX Disk BASIC toevoegt.
Wilt u MSX-DOS toch opstarten zonder het commando CALL SYSTEM, dan kunt u gebruik maken van de warme of koude start zoals dat in hoofdstuk 2 werd besproken.

Samenvatting

In MSX Disk BASIC dienen alle filenamen te worden opgegeven tussen aanhalingstekens ('quotes'), terwijl onder MSX-DOS de filenaam zonder die aanhalingstekens dient te worden opgegeven.
Het volgende overzicht bevat de statements voor Disk BASIC met daarnaast het statement in MSX-DOS dat dezelfde functie heeft:

MSX Disk BASIC	*MSX-DOS*
CALL FORMAT	FORMAT
FILES	DIR
KILL	DEL of ERASE
NAME AS	REN of RENAME
COPY TO	COPY

Appendix E Scheidingstekens en alternatieve startprocedures

Scheidingstekens

In dit boek gebruikten we alleen de spatie als scheidingsteken bij MSX-DOS. Er kunnen ook andere scheidingstekens worden gebruikt, zie paragraaf 7.2.

Bij nagenoeg alle commando's is de TAB toegestaan als scheidingsteken omdat die bestaat uit meerdere spaties. Alleen achter het commando BASIC geven alle scheidingstekens een foutmelding behalve de spatie! Dus het gebruik van een TAB is hier niet mogelijk.

Bij het commando DATE zijn alle spaties en de TAB toegestaan, alle andere scheidingstekens geven een foutmelding.

Alternatieve startprocedures

Het kan gebeuren dat u in het normale MSX BASIC wilt werken zonder gebruik te maken van Disk BASIC. Het grootste voordeel is de winst in geheugenruimte. Een eventueel programma dient u dan vanaf de cassetterecorder te laden.
Indien u na het aanzetten (of het indrukken van de resetknop) de SHIFT-toets houdt ingedrukt tot het versienummer in de linker bovenhoek van het beeldscherm verschijnt, staat de computer in MSX BASIC. De vrije geheugenruimte is dan 28.815 bytes, ongeveer 4 Kbytes meer dan de normale opstartprocedure voor MSX-DOS of Disk BASIC. Met deze procedure wordt de diskette-interface namelijk niet herkend.
De interface reserveert twee buffers voor de diskdrives en reserveert een deel van het geheugen voor de specifieke MSX Disk BASIC-commando's, vandaar de besparing van ca. 4 Kbytes.

De volgende opstartprocedure schakelt het buffergeheugen van diskdrive B uit, waardoor er ca. 1 Kbyte aan geheugenruimte wordt gewonnen ten opzichte van het normale Disk BASIC.
U dient nu de <CONTROL>-toets ingedrukt te houden tijdens het opstarten. Bevinden de twee MSX-DOS-systeemfiles zich op de geplaatste diskette, dan wordt er normaal opgestart in MSX-DOS.
Het specificeren van diskdrive B, rechtstreeks of via commando's, zal leiden tot een foutmelding.
U heeft nu een vrije geheugenruimte van 25.502 bytes. Normaal heeft men een capaciteit van 24.456 bytes.

Deze laatste procedure werkt alleen als er één diskdrive is aangesloten. Staan er twee diskdrives aangesloten, dan zal de pro-

cedure niet werken en zal de computer op de normale manier opstarten.

Appendix F Voorbeeldprogramma's

Telefoonboek

Dit programma vormt een eenvoudig telefoonboek. Het programma maakt voor de opslag van de gegevens gebruik van een sequentieel bestand. In het programma zien we duidelijk de modulaire opbouw, waardoor er veel subroutines worden aangeroepen. Het voordeel van een dergelijke programma-opbouw is de geheugenruimte die het programma in beslag neemt, deze is minimaal.

Er is bewust gekozen voor het steeds opnieuw invoeren van een bestandsnaam, hierdoor kan men snel zoeken in verschillende bestanden. Alle bestanden krijgen bij het openen van een nieuw bestand de extension '.DAT' van 'data'.

```
10 CLS
20 CLEAR 600
40 ON ERROR GOTO 440
50 LOCATE 5,1:PRINT"*** TELEFOONBOEK ***"
60 LOCATE 7,4:PRINT"*** KEUZEMENU ***"
70 LOCATE 5,6:PRINT"Opzetten bestand   <1>"
80 LOCATE 5,8:PRINT"Toevoegen bestand  <2>"
90 LOCATE 5,10:PRINT"Zoeken in bestand  <3>"
110 LOCATE 5,12:PRINT"Einde programma    <4>"
120 LOCATE2,18:PRINT"uw keuze "
130 KN$=INKEY$:KN=VAL(KN$)
140 IF KN=1 THEN 170
150 IF KN >4 OR KN <2 THEN 130
160 ON KN GOTO 120,260,300,430
170 LOCATE 2,18:PRINT"Nieuw bestand "
180 KN$=INKEY$
190 IF KN$="j" OR KN$="J" THEN 220
200 IF KN$="n" OR KN$="N" THEN 10
210 GOTO 180
220 CLS:LOCATE 5,1:PRINT"*** Bestand maken ***"
230 GOSUB 460
240 OPEN BN$ FOR OUTPUT AS #1
250 GOSUB 520:GOSUB 540:CLOSE:GOTO 10
260 CLS:GOSUB 460
270 CLS:A$="Toevoegen"
280 OPEN BN$ FOR APPEND AS #1
290 GOSUB 520:GOSUB 540:CLOSE:GOTO 10
```

```
300 CLS:GOSUB 460
310 OPEN BN$ FOR INPUT AS #1
320 CLS:A$="Zoeken":GOSUB 520
330 LOCATE 3,6:PRINT SPC(20)"? alle namen"
340 LOCATE 5,7:INPUT "Naam:";NN$
350 IF NN$="?" THEN N=0:GOTO 370
360 N=1
370 GOSUB 650:IF N=0 THEN 390
380 IF LEFT$(NA$,(LEN(NN$)))=NN$ THEN 390 ELSE 370
390 T=T+1
400 LOCATE 2,9+T*2:PRINT NA$,TE$
410 IF T=5 THEN GOSUB 680
420 GOTO 370
430 CLOSE:CLS:LOCATE 12,12:PRINT"TOT ZIENS":END
440 CLOSE:CLS:LOCATE 2,1:PRINT"*** Fout gemaakt ***"
450 FOR I=1 TO 2000:NEXT:GOTO 10
460 LOCATE 5,3:PRINT SPC(32)
470 LOCATE 5,3:INPUT "Bestandsnaam ";BN$
480 IF LEN(BN$) >8 THEN 460
500 BN$="a:"+BN$+".dat"
510 RETURN
520 CLS:LOCATE 2,1:PRINT"*** ";A$;" in bestand ***"
530 RETURN
540 LOCATE 29,6:PRINT"* einde"
550 LOCATE 3,7:INPUT"Naam:";NA$
560 IF NA$="*" THEN RETURN
570 LOCATE 3,9:INPUT"Telefoon:";TE$
580 LOCATE 3,15:PRINT"Is dit correct"
590 KN$=INKEY$
600 IF KN$="j" OR KN$="J" THEN 630
610 IF KN$="n" OR KN$="N" THEN 640
620 GOTO 590
630 PRINT#1,NA$;",";TE$
640 GOSUB 520:GOTO 540
650 REM
660 IF EOF(1)=-1 THEN 740
670 INPUT#1,NA$,TE$:RETURN
680 LOCATE 2,21:PRINT"Volgende scherm"
690 KN$=INKEY$
700 IF KN$="n" OR KN$="N" THEN CLOSE:GOTO 10
710 IF KN$="j" OR KN$="J" THEN 730
720 GOTO 690
730 T=0:CLS:GOSUB 520
740 IF EOF(1)=-1 THEN LOCATE 2,19:PRINT
    "Er zijn geen gegevens meer":CLOSE:GOTO 680
750 RETURN
```

Uitbreiding telefoonboek

Dit programma is een uitbreiding op het vorige programma en bevat een sorteerroutine, die gebruik maakt van een extra bestand waarin de gesorteerde records tijdelijk worden weggeschreven. De eerste keer dat de sorteerroutine zijn werk gaat doen wordt er uit het oorspronkelijke bestand gelezen. Is er inderdaad gesorteerd, dan wordt bij de tweede keer dat de sorteerroutine gaat beginnen vanuit het gesorteerde bestand gelezen. Dit blijft om en om gaan tot het gehele bestand is gesorteerd. Ten slotte wordt in regel 1030 of 1040 de naam van het bestand aangepast.
Als u het vorige programma heeft ingetoetst, kunt u dat laden en daarna deze programmaregels invoeren.

```
30  MAXFILES=2
100 LOCATE 5,12:PRINT"Sorteren bestand   <4>"
110 LOCATE 5,14:PRINT"Einde programma    <5>"
150 IF KN >5 OR KN <2 THEN 130
160 ON KN GOTO 120,260,300,760,430
490 SN$="a:"+BN$+".sor"
760 CLS:LOCATE 5,0:PRINT"*** SORTEREN BESTAND ***"
770 GOSUB 460
780 A=7:R=R+1:Q=0
790 IF INT(R/2)=R/2 THEN A=9
800 IF A=9 THEN 840
810 OPEN BN$ FOR INPUT AS #1
820 OPEN SN$ FOR OUTPUT AS #2
830 GOTO 860
840 OPEN SN$ FOR INPUT AS #1
850 OPEN BN$ FOR OUTPUT AS #2
860 FOR I=1TOA
870 IF EOF(1)=-1THEN 920
880 INPUT#1,NA$,TE$
890 X$(I)=NA$:Y$(I)=TE$
900 NEXTI
910 P=0
920 FOR C=1TO I-2
930 IF X$(C)>X$(C+1)THEN SWAPX$(C),X$(C+1):SWAPY$(C
    Y$(C+1):P=P+1:Q=Q+1
940 NEXT C
950 IF P>0THEN 910
960 FOR X=1 TO I-1
970 PRINT#2,X$(X);",";Y$(X)
980 NEXT
990 ERASE X$,Y$
1000 IF EOF(1)<> -1 THEN 860
```

```
1010 CLOSE
1020 IF Q>1 THEN 780
1030 IF A=7 THEN KILL BN$:NAME SN$ AS BN$:GOTO 10
1040 KILL SN$:GOTO 10
```

Automatisch formatteren en kopiëren

Dit is een BATCH-programma dat werkt onder MSX-DOS. Hiermee kunt u een nieuwe diskette formatteren en voorzien van de MSX-DOS-systeemfiles.
U roept het programma op met het commando 'NIEUW'.

```
A>COPY CON NIEUW.BAT
REM Dit programma formatteert een
REM nieuwe diskette en voorziet
REM deze van de MSX-DOS-files
PAUSE
FORMAT
PAUSE MSX-DOS overzetten?
COPY A:COMMAND.COM B:
COPY A:MSXDOS.SYS B:
DIR B:
^Z
```

Variant op de eeuwig durende kalender

Ongetwijfeld het kortste voorbeeldprogramma dat u laat zien welke dag het is op een bepaalde datum. Dit programma werkt alleen onder MSX-DOS en wordt geactiveerd met het commando 'DATUM'.

```
A>COPY CON DATUM.BAT
DATE
DATE
DATUM
^Z
```

TREFWOORDENREGISTER